

**Registracija Vozila**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Marko Šljivić**

**Miloš Kosanović [Broj indeksa]**

**REr 24/17**

**[datum]**

**27.08.2020**

SADRŽAJ

[1. Uvod - 3 -](#_Toc512514140)

[2. Instalacija i podešavanje projekta - 3 -](#_Toc512514141)

[2.1. Instaliranje modula - 3 -](#_Toc512514142)

[3. Arhitektura aplikacije - 3 -](#_Toc512514143)

[3.1 Serverski deo - 4 -](#_Toc512514144)

[3.2 Klijentski deo - 4 -](#_Toc512514145)

[3.3 Baza podataka - 5 -](#_Toc512514146)

[3.4 Komunikacija - 5 -](#_Toc512514147)

[4. Rad aplikacije - 5 -](#_Toc512514148)

[4.1 Opis implementacije - 5 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 6 -](#_Toc512514150)

[5. Literatura - 9 -](#_Toc512514151)

# 1. Uvod

U ovom projektu obrađena je izrada veb aplikacije za pregled i objavljivanje recepata, kao i njihovo ocenjivanje. Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS sa Bootstrap framework-om, AnimateCSS i JavaScript dok su na serverskoj strani korišćeni NodeJS sa Express framework-om.Alati koji su korišćeni prilikom izrade aplikacije su Google Chrome pretraživač̌,Visual Studio Code kao text editori Intellij kao IDE .

U aplikaciji možemo lako i jednostavno registrovati vozilo ili videti cenu registracije vozila po želji. Možemo pristupiti stranici za pregled svih registrovanih vozila kao i stranici za najčešće registrovana vozila.

# 2. Instalacija i podešavanje projekta

Da bismo pokrenuli nodejs aplikaciju potrebno je da instaliramo NodeJs, *a za ovaj projekat* koristimo i MySQL bazu, pa je potrebno instalirati Wampp server i ubaciti bazu preko phpmyadmin panela.

## 2.1. Instaliranje modula

U ovom projektu nalazi se **package.json** fajl, to je fajl koji se inicijalizuje sa projektom, naravno na zahtev programera, i ukoliko popunimo sve informacije ispravno biće kreiran fajl. Zatim kad krenemo da instaliramo nove module i ukoliko upotrebimo neke ključne reči (-S , -save) prilikom instalacije u ovom fajlu biće nam upisani svi moduli koje koristimo za našu aplikaciju. U koliko je sve to ispravno kreirano, da bi smo na nekom drugom računaru pokrenuli i instalirali module, potreno je ukucati sledeću naredbu **npm install** ova naredba prvo pretražuje **package.json** fajl i u njemu traži i instalira sve dependence (tj. Module ili biblioteke) koji su potrebni za ovaj projekat.

# 3. Arhitektura aplikacije

Aplikacija u sebi sadrži korenski (engl. Root) direktorijum '/' koji u sebi sadrži '/node\_module' direktorijum koji sadrži module koji su potrebni NodeJS-u, '/public' direktorijum u kome je smešten klijentski deo aplikacije, kao i četiri datoteke '/package.json' i '/package-lock.json' koje predstavljaju JSON dokumenta koji opisuju samu aplikaciju i sadrže spisak modula od kojih je serverski deo aplikacije zavistan i app.js datoteka u kojoj je kod NodeJS servera. Svakoj datoteci iz '/public' direktorijuma je moguće pristupiti javno, odnosno direktno iz URL-a veb čitača. '/public' direktorijum se sastoji iz četiri foldera: images, gde se nalaze slike koje su korišćene za projekat, javascript gde se nalaze sve potrebne javascript datoteke, stylesheets, gde se nalaze sve potrebne css datoteteke i assets gde se skladište dodatni fajlovi koje smo implementirali u našem projektu. U routes folderu se nalazi naš backend odnosno sve nodejs javascript datoteke koje odgovaraju svakoj html stranici. U views folderu se nalazi frontend, ondosno sve ejs datoteke korišćene u ovom projektu.

## 3.1 Serverski deo

Na serverskoj straini koristimo node.js Express framework za upravljanje rutama, konektovanje sa bazom podataka, celokupna logika sajta napisana je u node.js-u po modelu MVC.

Node.js je serverska JavaScript platforma koja se sadrži od minimalne core biblioteke pored bogatog ekosistema. Radi na V8 JavaScript engine-u, što znači da je veoma brz u izvršavanju.

Express.js je veb aplikacija u okviru Node.js servera, dizajnirana za izradu јednostraničnih, višestraničnih i hibridnih veb aplikacija. To je de fakto standardni serverski okvir za Node.js.

Model-View-Controller (MVC) je arhitekturalni patern koji se koristi u razvoju softvera. U složenim aplikacijama koje prikazuju korisniku ogromne količine podataka programeri često žele da razdvoje kod koji se bavi podacima od onog koji se bavi interfejsom, tako da razvoj oba postane lakši i jednostavniji.

## 3.2 Klijentski deo

HTML odnosno HyperText Markup Language je opisni jezik pomoću koga se kreiraju sajtovi, tačnije funkcionalnost elemenata jedne web stranice. S druge strane, CSS (Cascading Style Sheets) predstavlja jezik koji se koristi za uređivanje samog izgleda i formatiranje svih elemenata koje vidimo na sajtu. Iako na prvi pogled možda deluje da HTML i CSS zapravo imaju istu funkciju, njihova namena je potpuno drugačija, što je doprinelo tome da se ova dva jezika idealno dopunjuju.

EJS je jednostavan obrazac za jezik koji vam omogućava da generišete HTML markiranje jednostavnim JavaScript-om. Nema religioznosti o tome kako organizovati stvari. Nema ponovnog otkrivanja iteracije i protoka kontrole.

Za dodatnu stilizaciju stranice kao i mogućstva za lakšim kreiranjem responsive stranice koristimo Bootstrap framework. On sadrži dosta predefinisanih funkcija za stilizovanje tako da dosta umanjuje rad sa našim css dokumentom i takođe omogućava da aplikacija izgleda savršeno na svim veličinama ekrana.

## 3.3 Baza podataka

MySQL je višenitni višekorisnički SQL sistem za upravljanje bazama podataka. Sistem radi kao server, obezbjeđujući višekorisnički interfejs za pristup bazi podataka.

## 3.4 Komunikacija

# 

# 4. Rad aplikacije

## 4.1 Opis implementacije

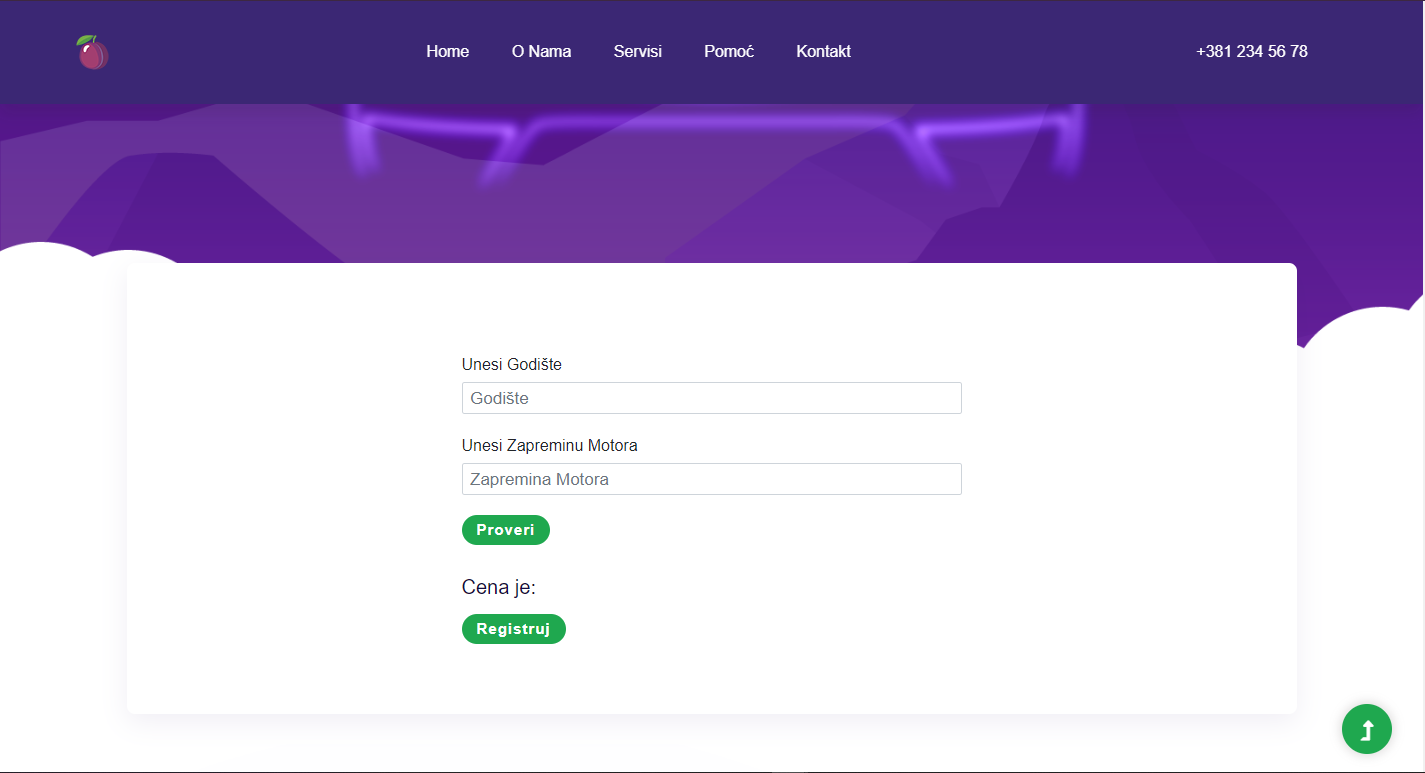
## 

## Ovaj deo koda je u okviru rute ‚registruj\_vozila/registuj‘ ovde se upisuje vozilo koje se registruje na sledecem nacinu:

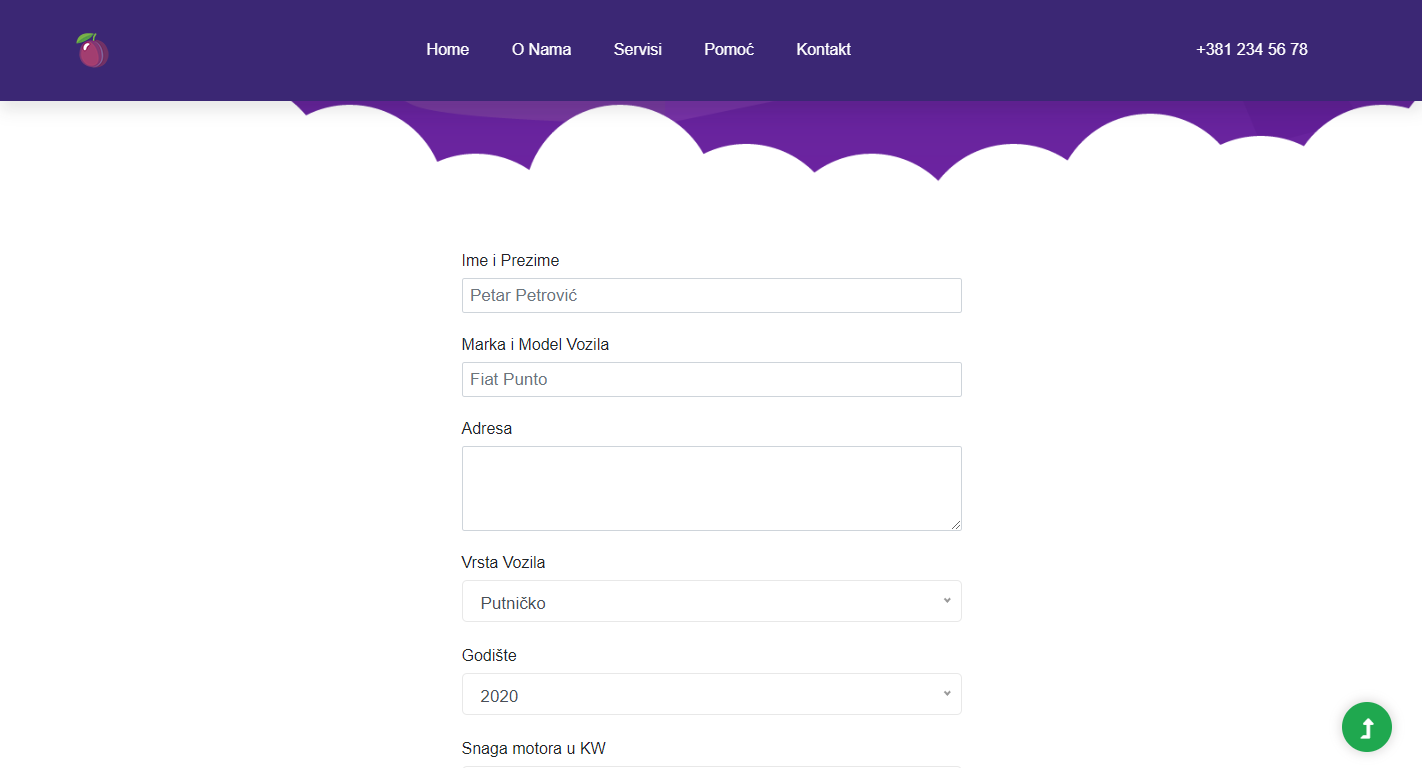
## Prvo sta radimo jeste dobijanje trenutnog datuma i radimo inkrementiranje godine.Zatim vidmo pozivanje funkcije cenaReg i smestane u promenljivu cena. CenaReg je funkcija koja na osnovu odredjenih uslova racuna cenu vozila na osnovu parametara koje smo joj poslali.Zatim imamo pisanje Query-a u koji radimo sql insert sto znaci da ce taj query da bude iskoriscen ako su svi uslovi ispunjeni. Zatim imamo query u kom proveravamo da li tablice koje su unetee za odredjeni auto vec postoji u bazi i da li je registrovana. Ako je ste korisnik dobija poruku da je vozilo registrovano do odredjenog datuma koji je upisan u bazi. Ako postoji a datum je prosao odradice se update query kojic e produziti datum registracije za godinu dana. Ako ne postoji broj tablice u bazi odradice se upis sa pomocu pomenutog insert query-a sql. Sql izvrsavamo uz pomoc funkcije pokrenisql.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

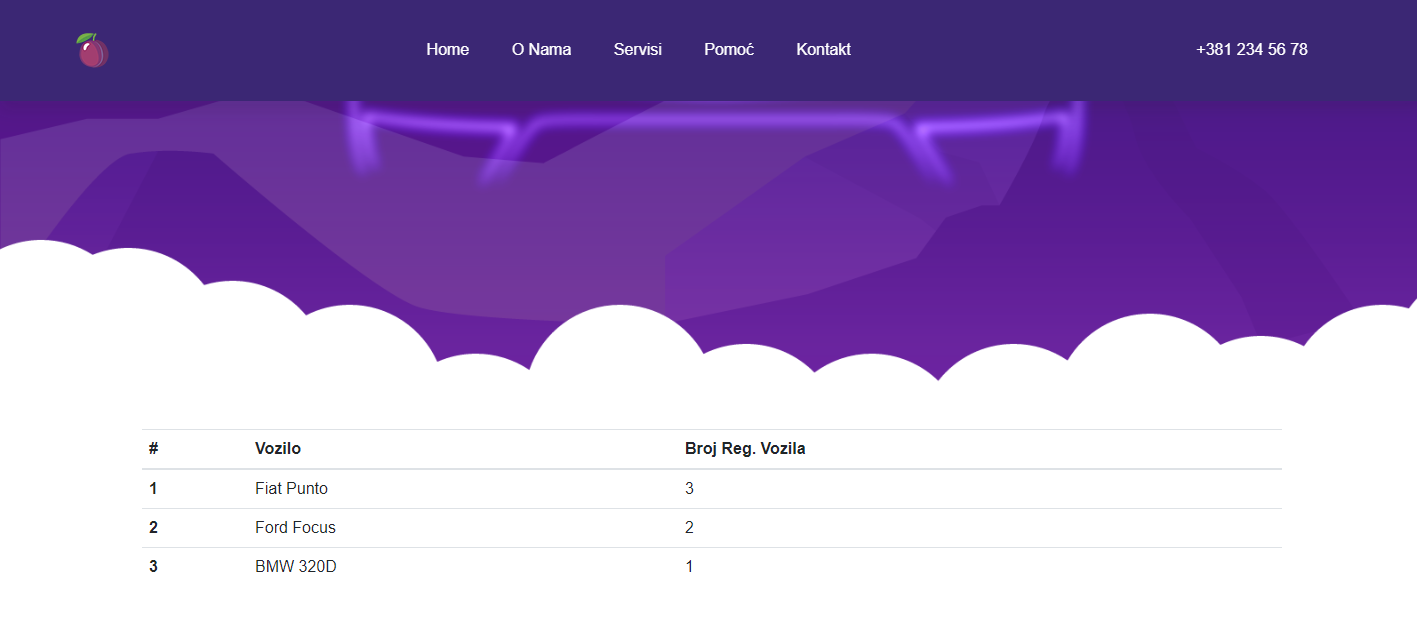
Nakon što pokrenemo naš projekat, server nas vodi na home stranicu gde možemo saznati sve informacije o sajtu kao i mini formu koja proverava cenu registracije na osnovu godišta i zapremine motora.



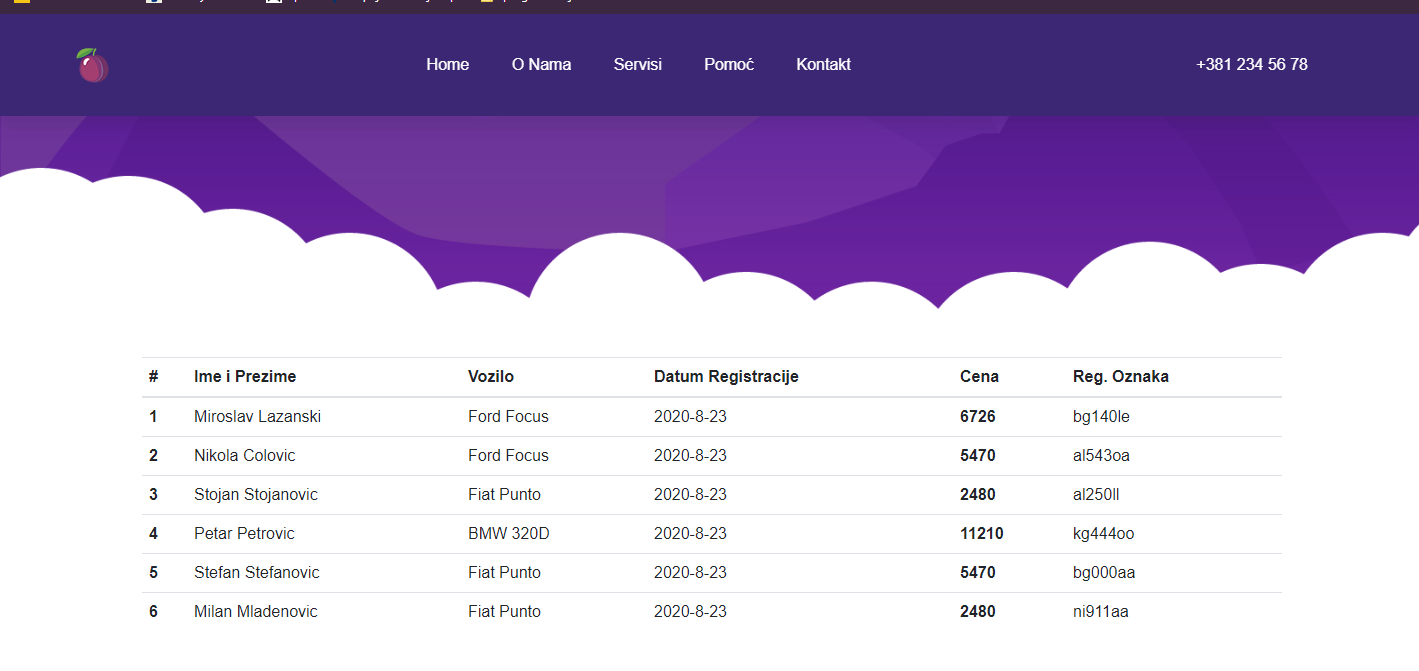
Ako je klijent zadovoljan cenom, klikom na dugme registruj, klijent će biti preusmeren na stranicu za registraciju sa popunjenim podacima koje je pre toga uneo.



Na stranici najčeše registrovani modeli, možemo videti sve modele koji su registrovani, kao i broj puta sortirani u opadajućem redosledu



Na stranici tabela registrovanih vozila možemo videti tabelu sa svim registrovanim vozilima, gde se nalazi ime i prezime vlasnika, marka i model vozila, datum registracije, cena i broj njegove registracije



# 

# 5. Literatura

[**https://stackoverflow.com/**](https://stackoverflow.com/)

[**https://www.youtube.com/**](https://www.youtube.com/)

[**https://github.com/**](https://github.com/)

[**https://nodejs.com/**](https://nodejs.com/)

[**https://w3.org/**](https://w3.org/)