Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



WEB auto – online evidencija prodatih automobila

Projektni rad iz predmeta:

Internet tehnologije

Profesor: Zorica Bogdanović Studenti:

Aleksa Ilić 67/16

Dalibor Jelićanin 164/16

Beograd, 2020.

Sadržaj

[Korisnički zahtev 1](#_Toc1819984)

[Opis tehnologija korišćenih u aplikaciji 2](#_Toc1819985)

[Bootstrap 2](#_Toc1819986)

[HTML5,CSS3 3](#_Toc1819987)

[JAVASCRIPT 4](#_Toc1819988)

[PHP 4](#_Toc1819989)

[AJAX 5](#_Toc1819990)

[SQL 7](#_Toc1819991)

[Chart.js 7](#_Toc1819992)

[Opis sistema 8](#_Toc1819993)

[UML dijagram klasa 8](#_Toc1819994)

[Arhitektura sistema – MVC 9](#_Toc1819995)

[Slučajevi korišćenja 10](#_Toc1819996)

[Dijagram slučaja korišćenja 10](#_Toc1819997)

[SK1 – Logovanje 11](#_Toc1819998)

[SK2 – Dodavanje automobila 11](#_Toc1819999)

[SK3 – Pretraga automobila 11](#_Toc1820000)

[SK4 – Izmena automobila 11](#_Toc1820001)

[SK5 – Brisanje automobila 12](#_Toc1820002)

[SK6 – Pregled automobila 12](#_Toc1820003)

[SK7 – Dodavanje prodaje 12](#_Toc1820004)

[SK8 – Pretraga prodaja 12](#_Toc1820005)

[SK9 – Pregled prodaja 12](#_Toc1820006)

[SK10 – Pregled statistike 13](#_Toc1820007)

[SK11 – Odjava 13](#_Toc1820008)

[Dijagram sekvenci 14](#_Toc1820009)

[DS1 – Logovanje 14](#_Toc1820010)

[DS2 – Dodavanje automobila 15](#_Toc1820011)

[DS3 – Pretraga automobila 16](#_Toc1820012)

[DS4 – Izmena automobila 17](#_Toc1820013)

[DS5 – Brisanje automobila 18](#_Toc1820014)

[DS6 – Pregled automobila 19](#_Toc1820015)

[DS7 – Dodavanje prodaje 19](#_Toc1820016)

[DS8 – Pretraga prodaja 20](#_Toc1820017)

[DS9 – Pregled prodaja 20](#_Toc1820018)

[DS10 – Pregled statistike 21](#_Toc1820019)

[DS11 – Odjava 21](#_Toc1820020)

[Dijagram komponenti 22](#_Toc1820021)

[Specifikacija REST API-a 23](#_Toc1820022)

[Prodati automobili po tipu automobila 23](#_Toc1820023)

[Broj prodaja po agentima 24](#_Toc1820024)

[Broj prodaja po mesecima 25](#_Toc1820025)

[Iznos prodaje po kompanijama 27](#_Toc1820026)

[Korisničko uputstvo 28](#_Toc1820027)

# 

# Korisnički zahtev

Potrebno je napraviti aplikaciju za beleženje prodaja automobila. Aplikacija se sastoji iz agentskog i administratorskog dela.

Administratorski deo sadrži funkcionalnosti za:

* Logovanje
* Dodavanje automobila
* Pretragu automobila
* Pregled automobila
* Izmenu I brisanje automobila
* Pregled automobila
* Pregled prodaja svih agenata
* Pregled statistike za izvršene prodaje svih agenata
* Odjavu sa sistema

Agentski deo sadrži funkcionalnosti za:

* Logovanje
* Dodavanje prodaje
* Pretragu prodaja ulogovanog agenta
* Pregled prodaja ulogovanog agenta
* Pregled statistike za izvršene prodaje ulogovanog agenta
* Odjavu sa sistema

Administrator i agent moraju se prijaviti na sistem kako bi koristili funkcionalnosti koje im njihove uloge pružaju.

# Opis tehnologija korišćenih u aplikaciji

## Bootstrap

Dva programera Mark Otto i Jacob Thornton rešila su da pomognu svim web dizajnerima i kreirali potpuno nov framework. Ova dva programera radila su za kompaniju Twitter i za potrebe ove kompanije napravili su odličan framework i rešili da ga podele sa celim svetom. Twitter Bootstrap je potpuno besplatan i što je najbolje on je open-source tako da svako može da ga dorađuje i prosledi drugima. Ovo nije samo običan framework već veoma moćan alat potreban svim web dizajnerima. Naime, Bootstrap može da pomogne da mnogo manje vremena bude potrošeno na pisanje koda, a da dizajn sajta bude veoma dobar.

Šta je zapravo Bootstrap?

Rešili ste da napravite sajt koji će imati „responsive“ dizajn, odnosno da se prilagođava uređaju, kako mobilnim telefonima tako tabletima i svim veličinama monitora. Znate i sami koliko bi vam vremena oduzelo da pravite posebne stilove za svaki od ovih uređaja i na kraju bi se odlučili eventualno za mobilne telefone i to one koji se najčešće koriste. Nakon toga tek dolaze problemi. Potrebno je sve to dizajnirati, napisati kod, proveriti ispravnost i izgubili ste jako puno vremena. Vi nemate vremena za sve ovo, potreban vam je Bootstrap.

Bootstap već ima sve ove stilove, na vama je samo da dodelite njegove klase vašim elementima i vaš sajt postaje „responsive“.

Na sajtu Bootstrap-a možete pronaći i dosta primera tako da vam neće biti teško da pronađete nazive klasa koji su vam potrebni. Sve što vam je potrebno jeste da pogledate kod u primeru.

Pored primera nalazi se dosta tutorijala gde možete naučiti kako Bootstrap funkcioniše, kao i osnove CSS-a i JavaScript-a koje su vam potrebne da biste što bolje uradili vaš sajt. Tutorijali su napisani tako da će i oni koji nikada nisu koristili CSS ili JavaScript moći lako da se snađu i naprave baš ono što žele.

Neki od sajtova koji koriste Bootstrap:

* + United Nations News Centre http://www.un.org/news/
  + Fat Dragon http://fatdragon.com.au/
  + Web appers http://www.webappers.com/
  + http://expo.getbootstrap.com/.

Prema rečima Mladena Plavšića , tipični razlozi koji ljude opredeljuju na to da ne koriste Twitter Bootstrap, ili bilo koji drugi framework, mogu se svrstati u nekoliko ključnih kategorija. Prevashodni faktor koji utiče na negativnu odluku je veličina, samim tim što produžava vreme odziva sajta koji koristi Bootstrap ili sličan framework. Ovaj problem se može prevazići korišćenjem Content Delivery Network-a (http://www.bootstrapcdn.com/) koji je besplatan i dovoljno brz. Pored toga, moguće je i uraditi custom download svih elemenata koji su potrebni iz frameworka bez preuzimanja celog paketa, što je takođe moguće uraditi na nekoliko načina, prvenstveno koristeći informacije sa sajta getbootstrap.com/customize/.

Drugi preovlađujući faktor koji utiče na timove ili freelancere da se ne upuštaju u bootstrapping je upravo uverenje da je sigurnije i bolje da se određene stvari razviju “kod kuće”. Preovladava mišljenje da dizajner ili developer najbolje zna kako da uradi projekat koji je pred njim, ali se onda postavlja pitanje da li, iako to programerima deluje najbolje, neko ko na kraju bude koristio proizvod deli to mišljenje. Možda je u pitanju trend ili jednostavna tendencija kopiranja u masovnoj proizvodnji, ali i pored toga postoji dovoljna sloboda da korišćenjem Bootstrapa budu realizovana izuzetno kvalitetna rešenja.

I pored mnogo drugih konkurentskih framework rešenja poput ZURB Foundation, Ink, Terrific itd., potrebno je uvek sagledati realne razloge zašto neki framework koristiti. Glavni kriterijumi pri izboru se svrstavaju u tri glavne kategorije, a to su dokumentacija, kvalitet i help&support koji je omogućen.

Tokom 2013. godine, pojavio se i Bootstrap v3 koji sadrži određene izmene u odnosu na pređašnje verzije (2.x). Postoje novine po pitanju imena klasa i uveden je novi grid, te nije u potpunosti “backwords compatible” sa pređašnjim verzijama, pa postoji mogućnost neintegracije nekih rešenja.

## HTML5,CSS3

**HTML** ([engl.](https://sr.wikipedia.org/wiki/%D0%95%D0%BD%D0%B3%D0%BB%D0%B5%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) ***H****yper****T****ext****M****arkup****L****anguage*, jezik za označavanje [hiperteksta](https://sr.wikipedia.org/wiki/%D0%A5%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82)) je opisni jezik specijalno namenjen opisu [veb stranica](https://sr.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1_%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).Pomoću njega se jednostavno mogu odvojiti elementi kao što su naslovi, [paragrafi](https://sr.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B3%D1%80%D0%B0%D1%84), [citati](https://sr.wikipedia.org/w/index.php?title=%D0%A6%D0%B8%D1%82%D0%B0%D1%82&action=edit&redlink=1) i slično. Pored toga, u HTML standard su ugrađeni elementi koji detaljnije opisuju sam dokument kao što su kratak opis dokumenta, ključne reči, podaci o autoru i slično. Ovi podaci su opštepoznati kao meta podaci i jasno su odvojeni od sadržaja dokumenta.

**CSS** ([engl.](https://sr.wikipedia.org/wiki/%D0%95%D0%BD%D0%B3%D0%BB%D0%B5%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA" \o "Engleski jezik) *Cascading Style Sheets*) je jezik formatiranja pomoću kog se definiše izgled elemenata veb-stranice. CSS sintaksa se sastoji od *opisa* izgleda elemenata u dokumentu. Opis može da definiše izgled više elemenata, i više opisa može da definiše jedan element. Na taj način se opisi *slažu* jedan preko drugog da bi definisali konačni izgled određenog elementa (otuda naziv *Cascading* ([engl.](https://sr.wikipedia.org/wiki/%D0%95%D0%BD%D0%B3%D0%BB%D0%B5%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) *cascade* -crep) da bi se dočaralo slaganje jednog stila preko drugog u definisanju konačnog izgleda elementa).

HTML tehnologiju u našoj aplikaciji primenili smo za prikaz sadržaja na stranicama. Kako sadržaj ne bi izgledao šturo i neorganizovano i kako bi podigli nivo interakcije sa korisnikom koristili smo i sve napredne mogućnosti CSS tehnologije. Za lepši prikaz sadržaja i adekvatnu stilizaciju istog CSS je korišćen na svim stranicama aplikacije.

## JAVASCRIPT

Java Script je najpopularniji jezik na Internetu, koji je dizajniran da poveća interaktivnost HTML strana. Razvila ga je kompanija Netscape Communications i uvela u Netscape Navigator, počevši od verzije 2.0, uporedo sa uvođenjem podrške za Javu. JavaScript je uveden radi povećanja interaktivnosti veb strana. Veb čitači koji podržavaju JavaScript interpretiraju kod.

**Javaskript** ([engl.](https://sr.wikipedia.org/wiki/%D0%95%D0%BD%D0%B3%D0%BB%D0%B5%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) *JavaScript*) je [skriptni](https://sr.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BD%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) [programski jezik](https://sr.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) koji se prvenstveno koristi za definisanje funkcionalnosti veb stranica na klijentskoj strani. Ono što je bitno zapamtiti jeste da JavaScript program može da se izvrašava samo u okviru WWW čitača i nigde drugde.

JavaScript je interpreterski jezik (što znači da se izvršava bez prethodnog kompajliranja).

Primenu JavaScript tehnologije izvršili smo na početnoj formi za logovanje korisnika gde smo na interaktivan način animirali elemente forme uz pomoć funkcija hide() i show(). Takođe, pomenute funkcije koristili smo za lepši prikaz obaveštenja koja korisnici dobijaju prilikom korišćenja aplikacije. JavaScript tehnologiju takođe smo primenili za prikaz brojača sekundi prilikom automatskog preusmeravanja na početnu stranicu aplikacije.

## PHP

*PHP: Hypertext Preprocessor* specijalizovani je [skriptni jezik](https://sr.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BD%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA" \o "Skriptni jezik) prvenstveno namenjen za izradu dinamičnog [veb](https://sr.wikipedia.org/wiki/%D0%A1%D0%B2%D0%B5%D1%82%D1%81%D0%BA%D0%B0_%D0%BC%D1%80%D0%B5%D0%B6%D0%B0) sadržaja i izvodi se na strani [servera](https://sr.wikipedia.org/wiki/Server" \o "Server).

Program koji se napiše u PHP-u ne zahteva [prevođenje](https://sr.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B0%D1%82%D0%BE%D1%80" \o "Kompilator) (kompajliranje), nego se [interpretira](https://sr.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80_(%D1%80%D0%B0%D1%87%D1%83%D0%BD%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%BE)" \o "Interpretator (računarstvo)) pri svakom izvršavanju. PHP interpretator može raditi po [PHP CGI](https://sr.wikipedia.org/w/index.php?title=Php_cgi&action=edit&redlink=1" \o "Php cgi (stranica ne postoji)) principu, odnosno tako što će interpretator postojati kao eksterna aplikacija, koja se poziva da izvrši datu skriptu svaki put kad bude zahtevana od nekog korisnika, a može biti instaliran i kao [modul](https://sr.wikipedia.org/w/index.php?title=%D0%9C%D0%BE%D0%B4%D1%83%D0%BB&action=edit&redlink=1" \o "Modul (stranica ne postoji)) veb-servisa. Druga varijanta je danas u najvećoj upotrebi jer pruža znatno veću brzinu izvršavanja - interpretator je na taj način uvek učitan u memoriju te se ne mora pozivati spoljašnji program.

Pomoću PHP-a se može kreirati HTML stranica na serveru pre njenog slanja na lokalni računar klijentu popunjenu dinamičkim sadržajem. Na ovaj način PHP kod koji je generisao stranu se ne može videti, već prikazuje čisti HTML kod. Početak svakog koda u programskom jeziku PHP potrebno je otvoriti tagom <?php i na kraju koda zatvoriti tagom ?>.

Sve promenljive u PHP programskom jeziku obeležavaju se sa znakom dolara na početku $. Pri izradi našeg projekta od ključnog značaja je korišćenje superglobalih varijabli pomoću kojih se realizuje slanje i primanje podataka iz neke forme. Dve superglobalne varijable koje smo koristili jesu get $\_GET["nazivpolja"] i post $\_POST["nazivpolja"].

Kao glavna tehnologija koju smo koristili pri izradi naše aplikacije, PHP nam je poslužio za izradu svake stranice koju naša aplikacija poseduje.Bitno je napomenuti da smo za upravljanje sesijom koristili globalnu promenljivu $\_SESSION. Uz pomoć ove promenljive omogućili smo prenos podataka o ulogovanom korisniku na svaku stranicu koju korisnik otvori sve do trenutka dok se ne izloguje iz aplikacije i time uništi sesiju. Pomoću md5 algoritma za enkripciju implementirali smo enkriptovan načina čuvanja lozinki korisnika u bazi podataka.

Takođe upotrebom PHP tehnologije realizovali smo implementaciju različitih korisničkih uloga sa različitim sadržajem i funkcionalnostima za korisnika i administratora.

Neke od osnovnih odrednica PHP – a su:

* PHP je akronim za Hypertext Preprocessor
* server – side skriptni programski jezik
* skripte se izvršavaju na serveru
* podržava rad sa različitim bazama podatama (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, itd.)
* open source softver (OSS)
* radi na različitim platformama (Windows, Linux, Unix i td.)
* kompatibilan je sa skoro svim serverima koji se danas koriste (Apache, IIS, itd. )
* lak je za učenje i efikasno funkcioniše na strani servera.

***Za PHP fajlove, bitne su sledeće činjenice:***

* mogu da sadrže tekst, HTML tagove i skripte
* vraćaju se klijentskom browser – u kao HTML tekst (plain HTML)
* imaju ekstenziju: “.php”, “.php3” ili “.phtml”

## AJAX

AJAX predstavlja web razvojnu tehniku za kreiranje interaktivnih web sadržaja. AJAX tehnologija se zasniva na razmeni malih količina podataka sa serverom, kako korisnik ne bi stalno morao da osvežava stranicu. Na taj način se omogućava potpuna interaktivnost, efikasnost i poboljšava funkcionisanje stranice uopšte. AJAX nije tehnologija sam po sebi, već termin koji se odnosi na korišćenje grupe tehnologija.

AJAX obuhvata:

* prezentaciju baziranu na standardima koristeći XHTML i CSS;
* dinamički prikaz i interakciju preko Document Object Model-a;
* razmenu i manipulaciju nad podacima koristeći XML i XSLT;
* asinhrono prikupljanje podataka uz pomoć XMLHttpRequest-a koji čini jezgro;
* i JavaScript koji povezuje sve u celinu.

Ideja koja se krije iza AJAX-a jeste da se stranica na kojoj se odvija web aplikacija učita samo jednom, a da se svaka dalja komunikacija sa serverom sakrije od očiju korisnika i obavlja bez ponovnog učitavanja čitave stranice. Svaki prenos podataka između servera i klijenta (u slučaju AJAX-a to je browser) vrši se u pozadini. Jasno je da je ovo nemoguće izvesti statičkim HTML-om, pa tu na scenu stupa JavaScript. JavaScript je zadužen za komuniciranje sa serverom – slanje HTTP zahteva, prijem podataka sa servera i njihov prikaz na stranici, a i interakcija sa korisnikom postiže se korišćenjem ovog jezika.

Neki od osnovnih principa na kojima bi trebalo da se zasnivaju AJAX aplikacije:

* Minimalan protok - AJAX aplikacije bi trebalo da šalju i primaju sa servera što je moguće manje informacija
* AJAX aplikacije predstavljaju korisniku drugačije modele interakcije od tradicionalnih web aplikacija. Nasuprot “klik – čekaj“ prirodi standardnih web aplikacija, AJAX predstavlja modele interakcije koji su slični desktop aplikacijama. Bez obzira koji se model interakcije koristi, najbitnije je obezbediti konzistentnost kako bi korisnik znao šta treba sledeće da uradi.
* Izbegavaju se nepotrebni elementi na strani kao što su animacije ili delovi koji trepere.
* Izbegava se preuzimanje cele strane - celokupna komunikacija sa serverom, nakon učitavanja inicijalne web strane treba da bude zadatak AJAX engine –a

AJAX omogućava različite funkcionalnosti:

* Validacija podatka u realnom vremenu - Podaci na formama kao što su: user IDs, serial numbers, postal codes, i ostali koji zahtevaju validaciju od strane servera, mogu biti provereni pre submit- ovanja forme.
* Autocompletion i autosuggest - Određeni delovi podataka koji se unose od strane korisnika(posetioca): e-mail adresa, ime, grad i ostali se mogu automatski kompletirati, tj. dovršiti odmah nakon što se unesu samo početni delovi.
* Učitavanje na zahtev - U skladu sa iniciranim događajem od strane korisnika, HTML strana može povući veću količinu podataka u pozadini i omogućiti učitavanje kompletne strane mnogo brže.
* Sofisticiran korisnički interfejs, kontrole i efekti – Kontrole kao što su meniji, tabele podatka, tekst editori, kalendari, status bar-ovi I sl. Omogućavaju korisnicima bolju interakciju, bez potrebe za učitavanjem kompletne stranice.
* Osvežavanje podataka - HTML stranice povlače podatke sa servera i omogućavaju najsvežije podatke kao što su rezultati, cene akcija, vreme ili neke druge podatke iz specifičnih aplikacija.
* Delimičan submit – bez potrebe za učitavanjem cele forme.
* Stranica kao aplikacija – web stranice postaju slične desktop aplikacijama

## SQL

SQL je standardni programski jezik za pristup i manipulaciju podacima iz relacionih baza podataka i podržan je od strane svih servera relacionih baza podataka. U relacionim bazama podaci su skladišteni preko niza tabela. Svaka tabela sadrži kolone koje opsiuju atribute podataka, a svaki red je instanca podataka.

Relational Data Base Management Systems (RDBMS), odnosno sistemi za upravljanje relacionim bazama podataka su se u praksi pokazali kao najbolji način upravljanja podacima.

**Chart.js**

Chart.js je jednostavna i fleksibilna JavaScript biblioteka za dizajnere i programere. Ovaj projekat je otvorenog koda (open source) koji održava zajednica. Koristi se za vizuelizaciju svojih podataka na različite načine, svaki od njih je animiran i prilagodljiv. Ima odlične performanse u svim modernim pretraživačima i responzivan je. Lako je započeti rad sa Chart.js-om, sve što je potrebno je da biblioteka bude uključena na vašoj stranici zajedno sa jednim <canvas> čvorom da bi se prikazao grafikon. Na svakom grafikonu imamo mogućnost kastomizacije. Neke od osnovnih opcija koje se koriste u u ovoj biblioteci su:

type – biramo vrstu grafikona koji zelimo da prikazemo (line, bar, radar, pie,…)

data – podaci koje zelimo da vizuelizujemo

datasets – skupovi podataka sa svojim opcijama i vrednostima

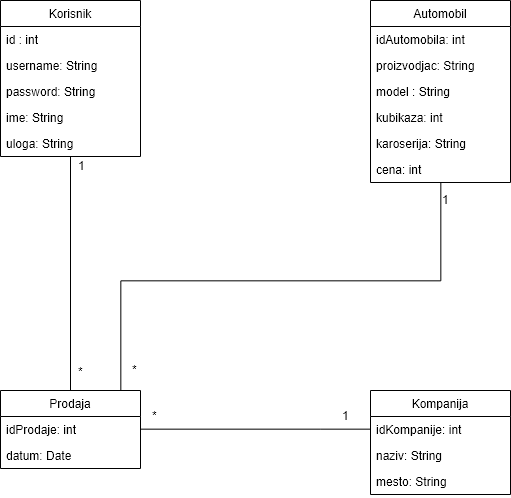
label - nazivi podataka

options – opcije za kastomizaciju grafikona

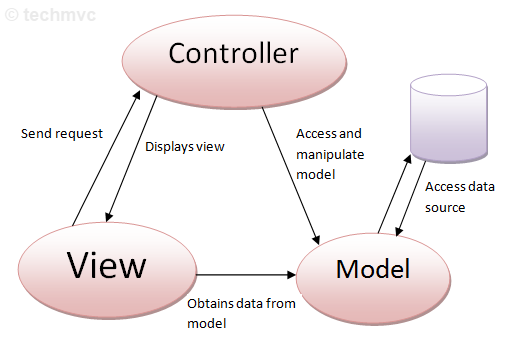
title – opcije za naslov grafikona

# Opis sistema

## UML dijagram klasa



# Arhitektura sistema – MVC



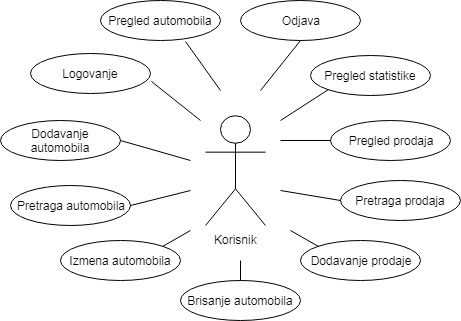
**Model** – Predstavlja strukture podataka. Klase u model sloju uglavnom služe za čitanje, dodavanje i ažuriranje podataka u bazi podataka.

**View** – Informacije prezentovane korisniku.

**Controller**  – handler - Služi kao posrednik između model-a, view-a  i ostalih resursa potrebnih za procesiranje HTTP zahteva.

# Slučajevi korišćenja

## Dijagram slučaja korišćenja



## SK1 – Logovanje

**Naziv:** Logovanje

**Akteri:** Korisnik

**Preduslov:** Korisnik postoji u bazi

**Osnovni scenario:**

1. Korisnik navigira do stranice za logovanje
2. U odgovarajuća polja unosi username i lozinku i potvrđuje unos pritiskom na dugme
3. Sistem prihvata prijavu korisnika i dozvoljava mu pristup aplikaciji

**Alternativni scenario:**

3.1 Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o tome.

3.2 Ukoliko u bazi ne postoji registrovani korisnik sa datim korisničkim imenom i lozinkom sistem obaveštava korisnika o tome.

## SK2 – Dodavanje automobila

**Naziv:** Dodavanje automobila

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela.
2. Administrator upisuje podatke o novom automobilu i pritiskom na dugme potvrđuje unos.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o tome.

## SK3 – Pretraga automobila

**Naziv:** Pretraga automobila

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator navigira do stranice za upravljanje automobilima.
2. Administrator unosi podatke za pretragu u obliku teksta i pritiskom na dugme izvršava pretragu.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK4 – Izmena automobila

**Naziv:** Izmena automobila

**Akteri:** Administrator

**Osnovni scenario:**

1. Adminstrator vrši pretragu automobila i pritiska dugme “izmeni” za izabrani automobil.
2. Administrator unosi podatke i pritiskom na dugme potvrđuje nove podatke o automobilu.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK5 – Brisanje automobila

**Naziv:** Brisanje automobila

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator vrši pretragu automobila.
2. Administrator bira automobil I pritiska dugme za brisanje.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK6 – Pregled automobila

**Naziv:** Pregled automobila

**Akteri:**  Administrator

**Osnovni scenario:**

1. Administrator navigira do stranice za upravljanje automobilima.
2. Administrator pregleda listu automobila.

**Alternativni scenario:**

2.1 Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK7 – Dodavanje prodaje

**Naziv:** Dodavanje prodaje

**Akteri:** Agent

**Osnovni scenario:**

1. Agent navigira do agentskog panela.
2. Agent upisuje podatke o novoj prodaji i pritiskom na dugme potvrđuje unos.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o tome.

## SK8 – Pretraga prodaja

**Naziv:** Pretraga prodaja

**Akteri:** Agent

**Osnovni scenario:**

1. Agent navigira do stranice za upravljanje prodajama.
2. Agent unosi podatke za pretragu u obliku teksta i pritiskom na dugme izvršava pretragu.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK9 – Pregled prodaja

**Naziv:** Pregled prodaja

**Akteri:**  Administrator, agent

**Osnovni scenario:**

1. Administrator/agent navigira do stranice za upravljanje prodajama.
2. Administrator/agent pregleda listu prodaja.

**Alternativni scenario:**

2.1 Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK10 – Pregled statistike

**Naziv:** Pregled statistike

**Akteri:**  Administrator, agent

**Osnovni scenario:**

1. Administrator/agent navigira do stranice za upravljanje statistikom.
2. Administrator/agent pregleda statistiku.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

## SK11 – Odjava

**Naziv:** Odjava

**Akteri:** Korisnik

**Osnovni scenario:**

1. Korisnik navigira do stranice za odjavu sa sistema.
2. Korisnik se pritiskom na dugme odjavljuje sa sistema.

**Alternativni scenario:**

* 1. Ukoliko sistem ne može da se poveže na bazu prikazuje poruku o datom problemu.

# Dijagram sekvenci

## DS1 – Logovanje

**Osnovni scenario:**

1. Korisnik navigira do stranice za logovanje
2. U odgovarajuća polja unosi email i lozinku i potvrđuje unos pritiskom na dugme
3. Sistem prihvata prijavu korisnika i dozvoljava mu pristup stranama



UnosUsername

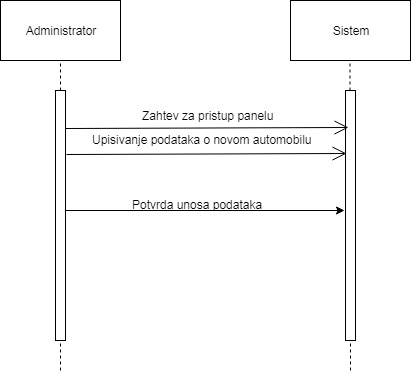
UnosUsername

## DS2 – Dodavanje automobila

UnosUsername

**Osnovni scenario:**

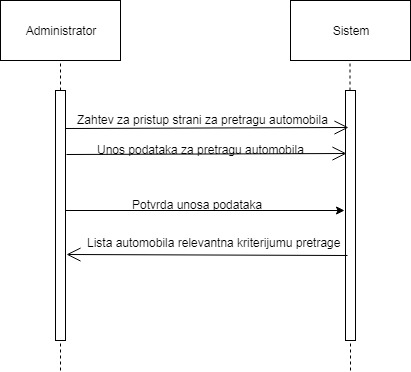
1. Administrator navigira do administratorskog panela.
2. Administrator upisuje podatke o novom automobilu i pritiskom na dugme potvrđuje unos.



## DS3 – Pretraga automobila

**Osnovni scenario:**

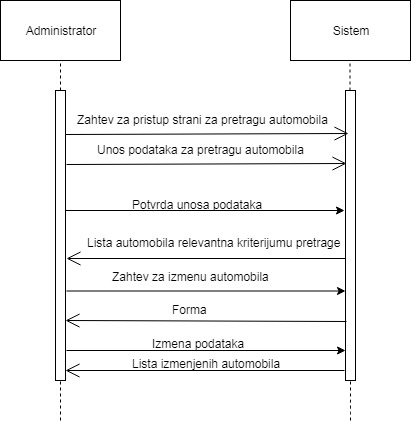
1. Administrator navigira do stranice za upravljanje automobilima.
2. Administrator unosi podatke za pretragu u obliku teksta i pritiskom na dugme izvršava pretragu.



## DS4 – Izmena automobila

**Osnovni scenario:**

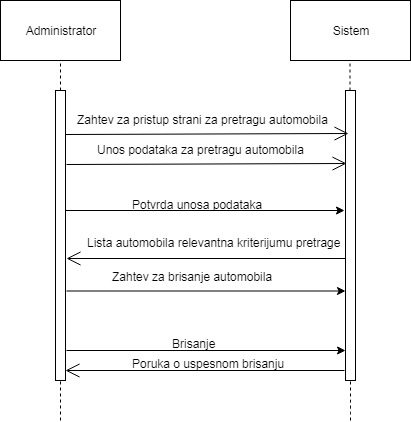
1. Adminstrator vrši pretragu automobila i pritiska dugme “izmeni” za izabrani automobil.
2. Administrator unosi podatke i pritiskom na dugme potvrđuje nove podatke o automobilu.



## DS5 – Brisanje automobila

**Osnovni scenario:**

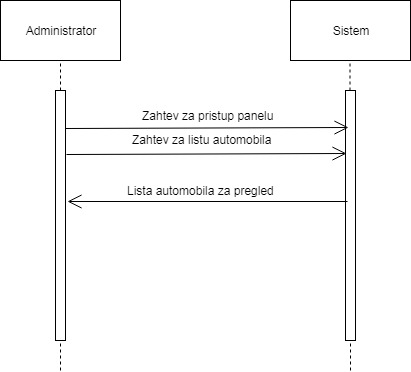
1. Administrator vrši pretragu automobila.
2. Administrator bira automobil I pritiska dugme za brisanje.



## DS6 – Pregled automobila

**Osnovni scenario:**

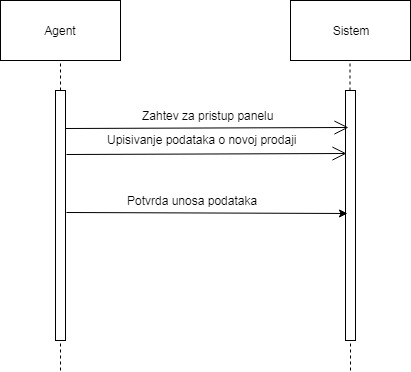
1. Administrator navigira do stranice za upravljanje automobilima.
2. Administrator pregleda listu automobila.



## DS7 – Dodavanje prodaje

**Osnovni scenario:**

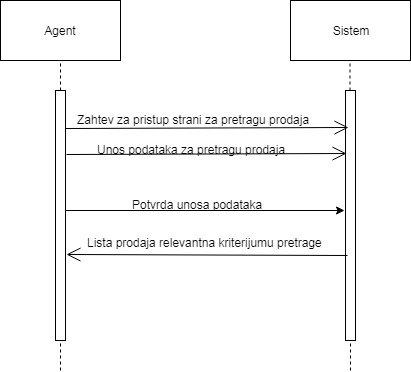
1. Agent navigira do agentskog panela.
2. Agent upisuje podatke o novoj prodaji i pritiskom na dugme potvrđuje unos.



## DS8 – Pretraga prodaja

**Osnovni scenario:**

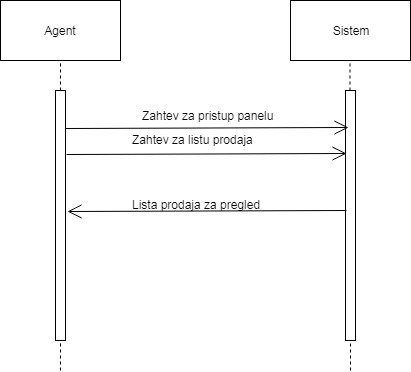
1. Agent navigira do stranice za upravljanje prodajama.
2. Agent unosi podatke za pretragu u obliku teksta i pritiskom na dugme izvršava pretragu.



## DS9 – Pregled prodaja

**Osnovni scenario:**

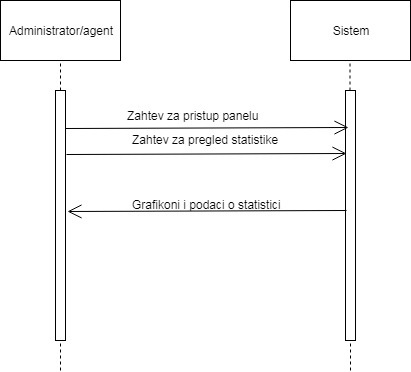
1. Administrator/agent navigira do stranice za upravljanje prodajama.
2. Administrator/agent pregleda listu prodaja.



## DS10 – Pregled statistike

**Osnovni scenario:**

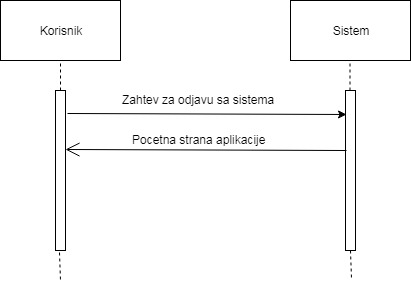
1. Administrator/agent navigira do stranice za upravljanje statistikom.
2. Administrator/agent pregleda statistiku.



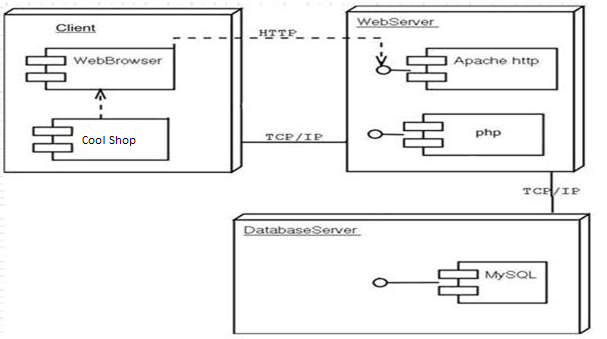
## DS11 – Odjava

**Osnovni scenario:**

1. Korisnik navigira do stranice za odjavu sa sistema.
2. Korisnik se pritiskom na dugme odjavljuje sa sistema.



# Dijagram komponenti



WEB Auto

# Specifikacija REST API-a

* **GET brojKupljenihTipovaAutomobila.php**
* **GET brojProdajaPoAgentima.php**
* **GET brojProdajaPoMesecima.php**
* **GET iznosProdajePoKompanijama.php**

## Prodati automobili po tipu automobila

|  |  |
| --- | --- |
| Opis funkcije | Pregled svih automobila |
| HTTP metoda | GET |
| URL | http://localhost/iteh/stats/brojKupljenihTipovaAutomobila.php |
| URL parametri | (nema) |
| HTTP body parametric | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata.  Primer: [      {         **"karoserija"**:"dzip",       **"brojKaroserija"**:"5"    },    {         **"karoserija"**:"hatchback",       **"brojKaroserija"**:"8"    },    {         **"karoserija"**:"karavan",       **"brojKaroserija"**:"5"    },    {         **"karoserija"**:"kupe",       **"brojKaroserija"**:"3"    },    {         **"karoserija"**:"limuzina",       **"brojKaroserija"**:"27"    },    {         **"karoserija"**:"pickup",       **"brojKaroserija"**:"6"    } ] |
| Format izlaznih parametara | application/json |

## Broj prodaja po agentima

|  |  |
| --- | --- |
| Opis funkcije | Lista automobila određenog tipa |
| HTTP metoda | GET |
| URL | http://localhost/iteh/stats/brojProdajaPoAgentima.php |
| URL parametri | (nema) |
| HTTP body parametric | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata.  Primer:  [      {         **"ime"**:"Dalibor Jelcanin",       **"brojProdaja"**:"26"    },    {         **"ime"**:"Aleksa Ilic",       **"brojProdaja"**:"28"    } ] |
| Format izlaznih parametara | application/json |

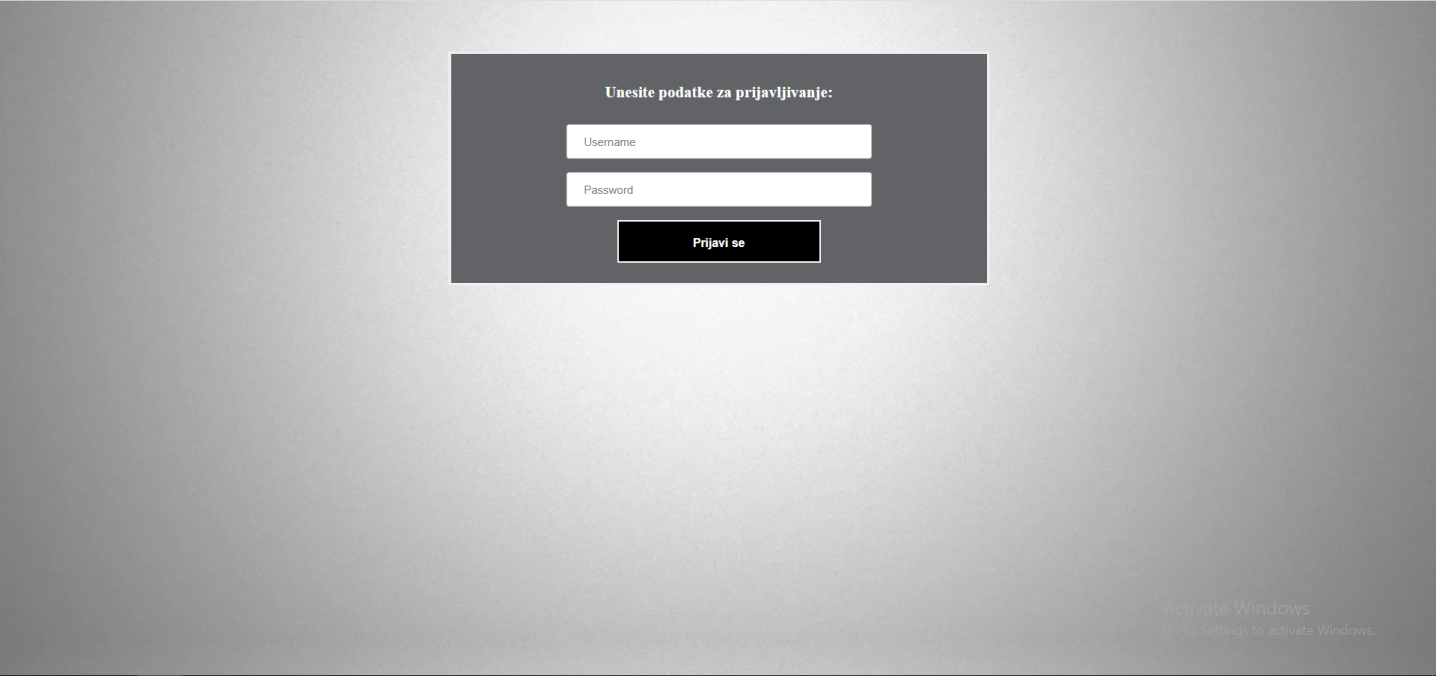
## Broj prodaja po mesecima

|  |  |
| --- | --- |
| Opis funkcije | Podaci o uređaju |
| HTTP metoda | GET |
| URL | http://localhost/iteh/stats/brojProdajaPoMesecima.php |
| URL parametri | (nema) |
| HTTP body parametric | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata.  Primer:  [      {         **"idAgenta"**:"1",       **"mesec"**:"1",       **"brojPoMesecu"**:"2"    },    {         **"idAgenta"**:"1",       **"mesec"**:"2",       **"brojPoMesecu"**:"4"    },    {         **"idAgenta"**:"1",       **"mesec"**:"3",       **"brojPoMesecu"**:"1"    },    {         **"idAgenta"**:"1",       **"mesec"**:"4",       **"brojPoMesecu"**:"1"    },    {         **"idAgenta"**:"1",       **"mesec"**:"5",       **"brojPoMesecu"**:"2"    },    {         **"idAgenta"**:"1",       **"mesec"**:"6",       **"brojPoMesecu"**:"1"    },    {         **"idAgenta"**:"1",       **"mesec"**:"7",       **"brojPoMesecu"**:"3"    },    {         **"idAgenta"**:"1",       **"mesec"**:"8",       **"brojPoMesecu"**:"1"    },    {         **"idAgenta"**:"1",       **"mesec"**:"9",       **"brojPoMesecu"**:"3"    },    {         **"idAgenta"**:"1",       **"mesec"**:"10",       **"brojPoMesecu"**:"2"    },    {         **"idAgenta"**:"1",       **"mesec"**:"11",       **"brojPoMesecu"**:"2"    },    {         **"idAgenta"**:"1",       **"mesec"**:"12",       **"brojPoMesecu"**:"4"    } ] |
| Format izlaznih parametara | application/json |

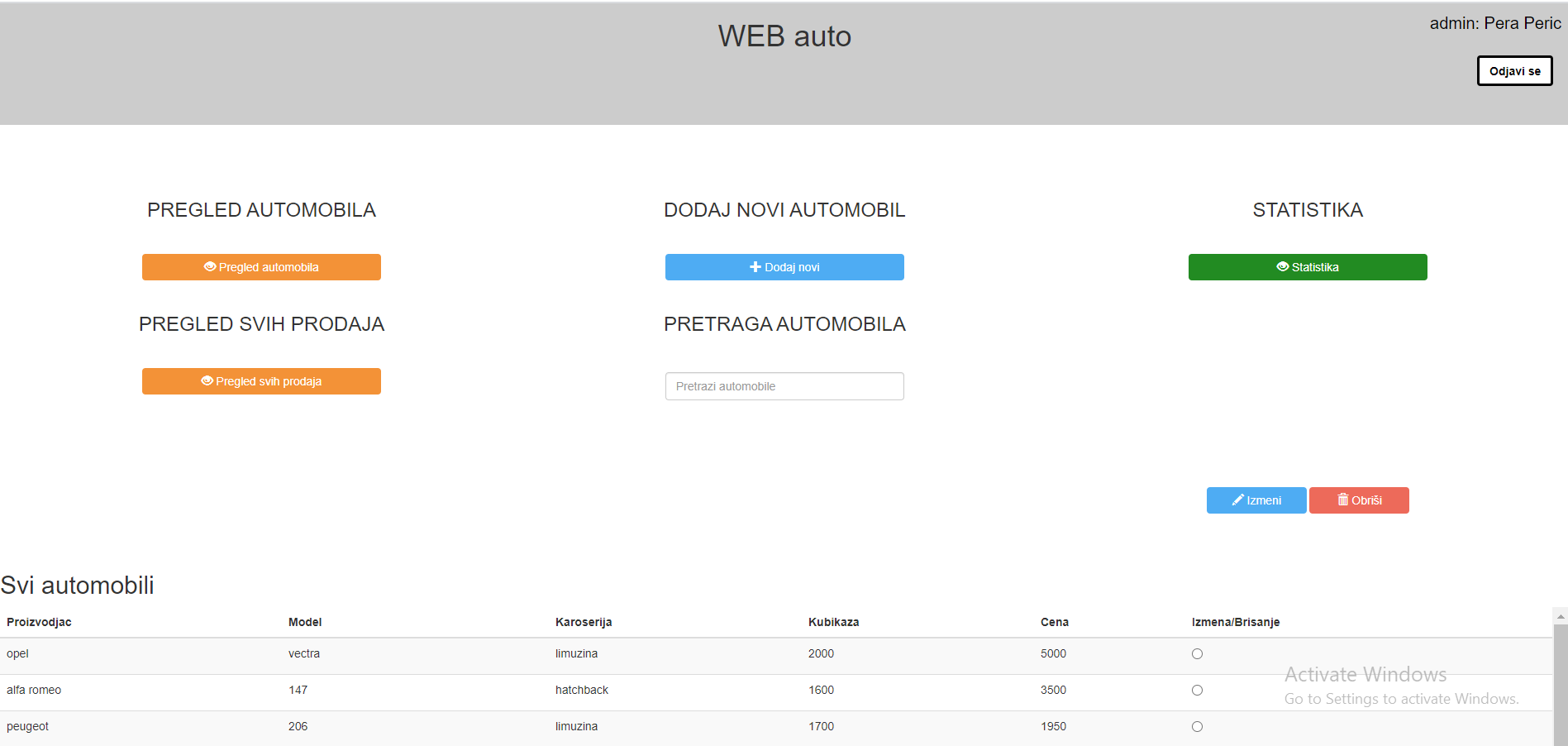
## Iznos prodaje po kompanijama

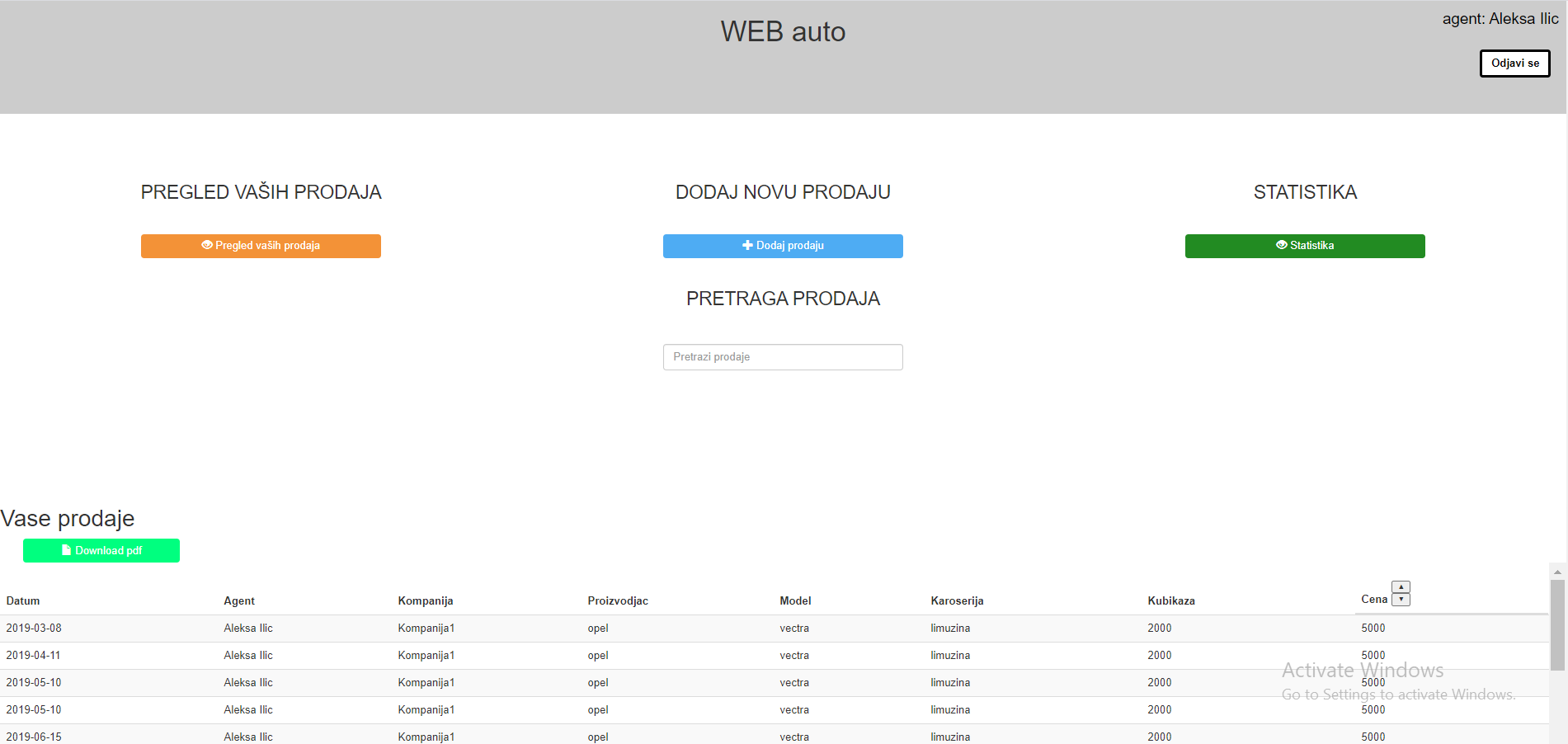
|  |  |
| --- | --- |
| Opis funkcije | Podaci o uređaju |
| HTTP metoda | GET |
| URL | http://localhost/iteh/stats/iznosProdajePoKompanijama.php |
| URL parametri | (nema) |
| HTTP body parametric | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametric | Niz JSON objekata.  Primer:  [      {         **"Naziv"**:"Kompanija1",       **"iznos"**:"100300"    },    {         **"Naziv"**:"Kompanija2",       **"iznos"**:"80850"    },    {         **"Naziv"**:"Kompanija3",       **"iznos"**:"77800"    } ] |
| Format izlaznih parametara | application/json |

**Korisničko uputstvo**

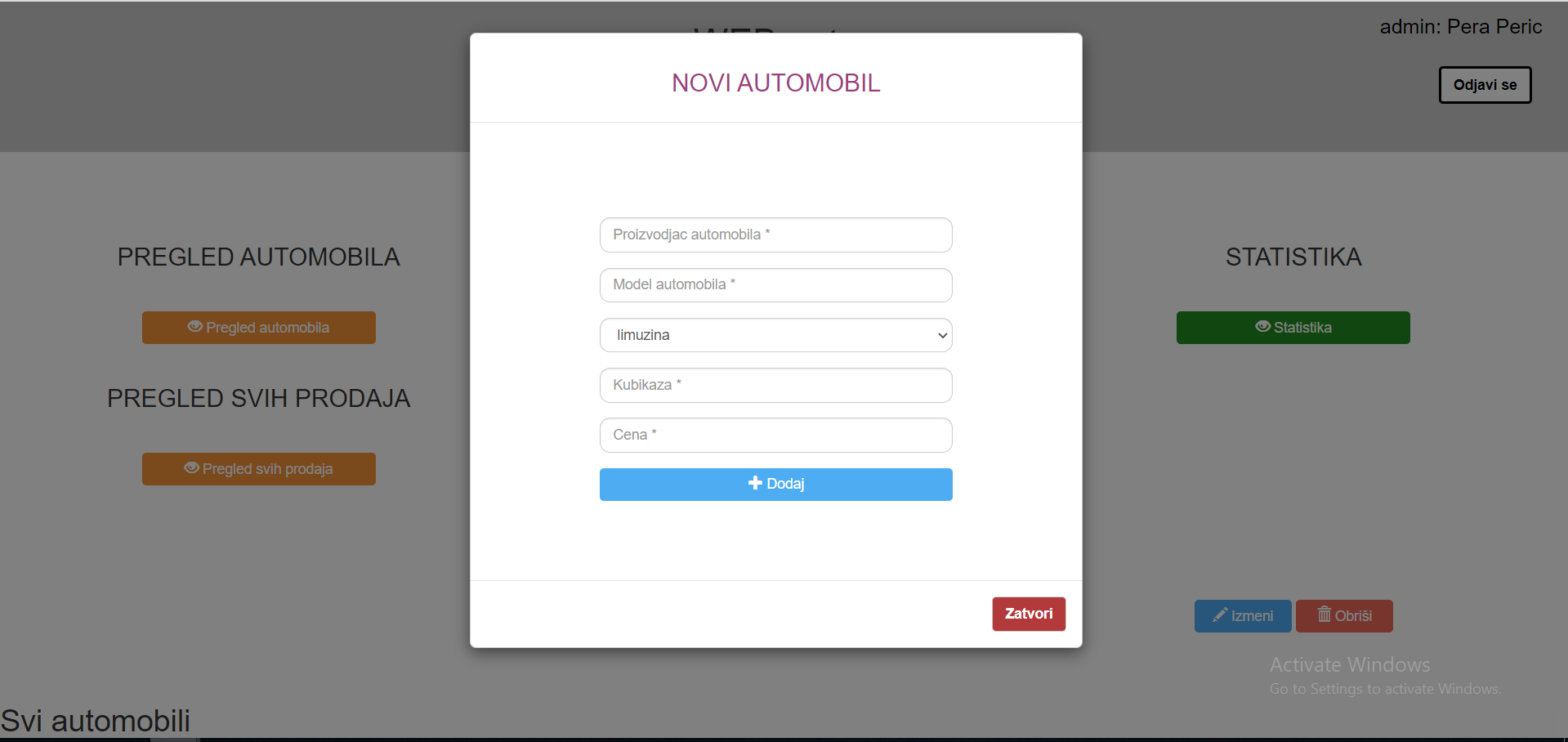
Pokretanjem aplikacije otvara se stranica za prijavljivanje.

Nakon uspesne prijava otvara se stranica čiji sadržaj zavisi od uloge korisnika.

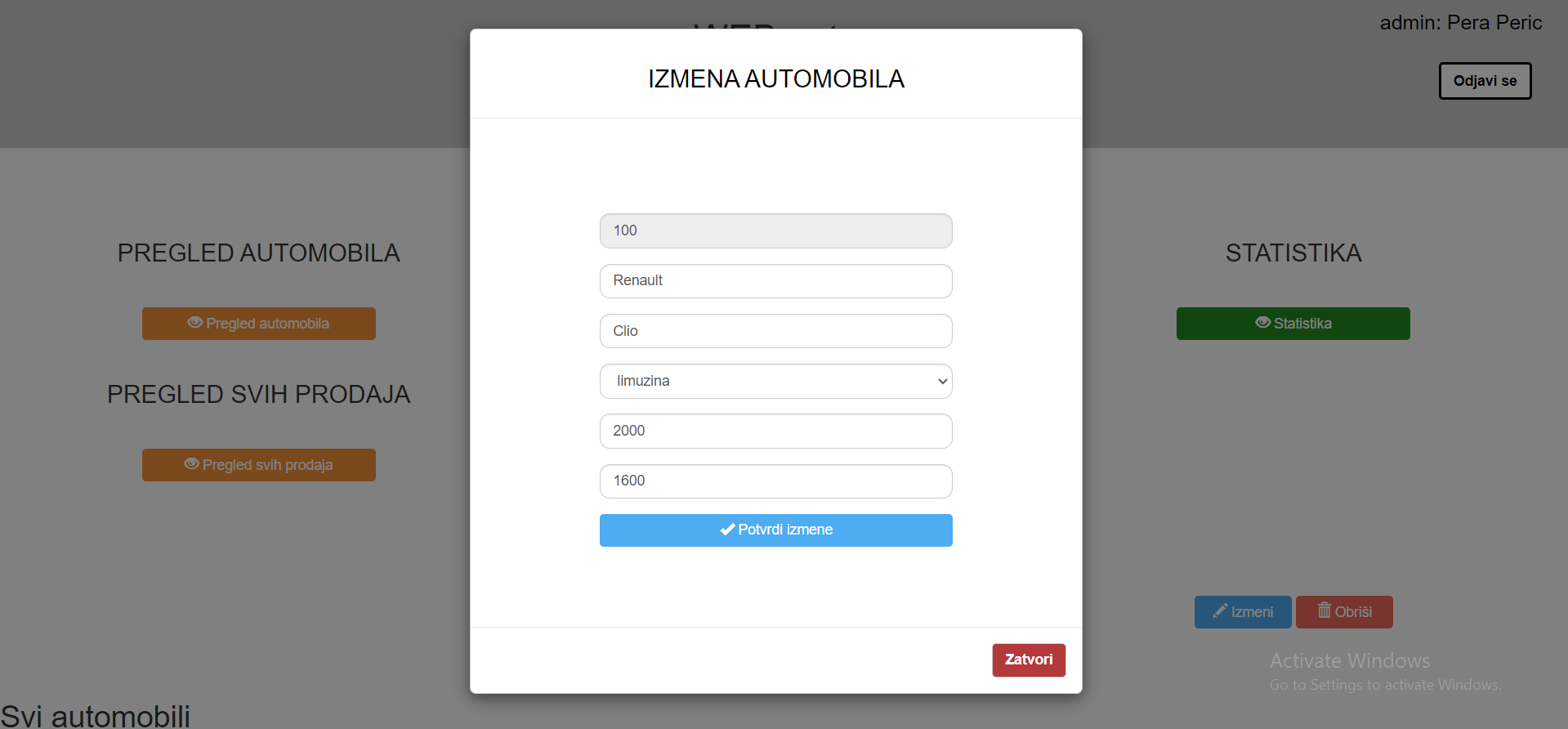




Administratori imaju mogućnost dodavanja novog automobila klikom na dugme dodaj novi, nakon čega se otvara forma za unos automobila.

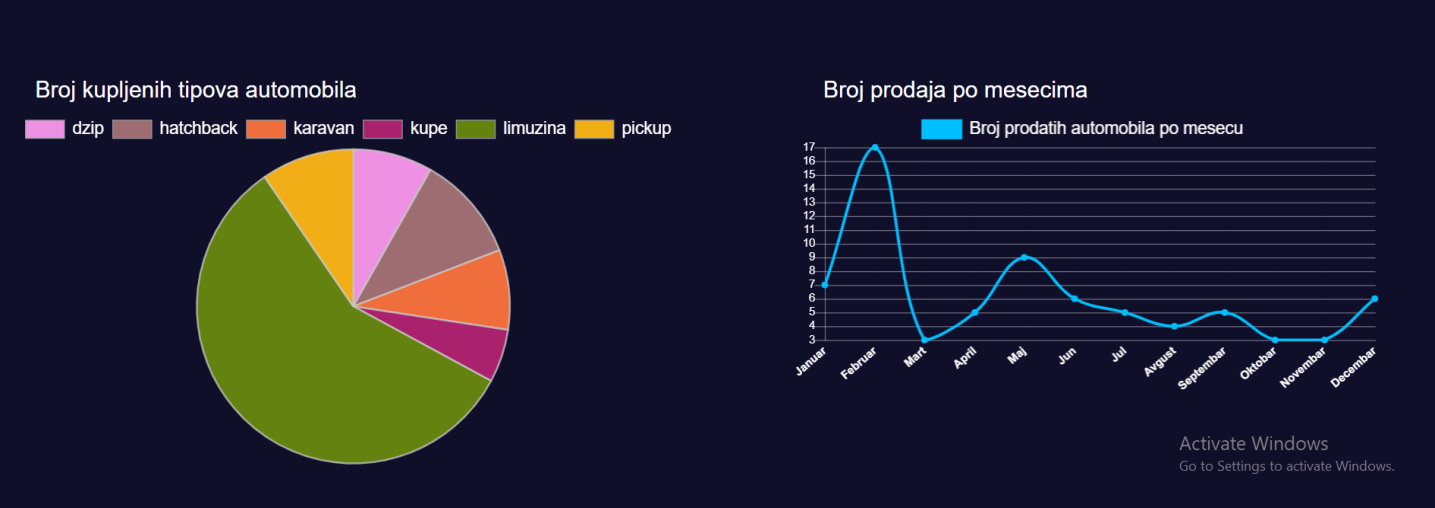


Izborom automobila iz tabele svih automobila, administratoru je omogućena izmena i brisanje izabranog automobila.

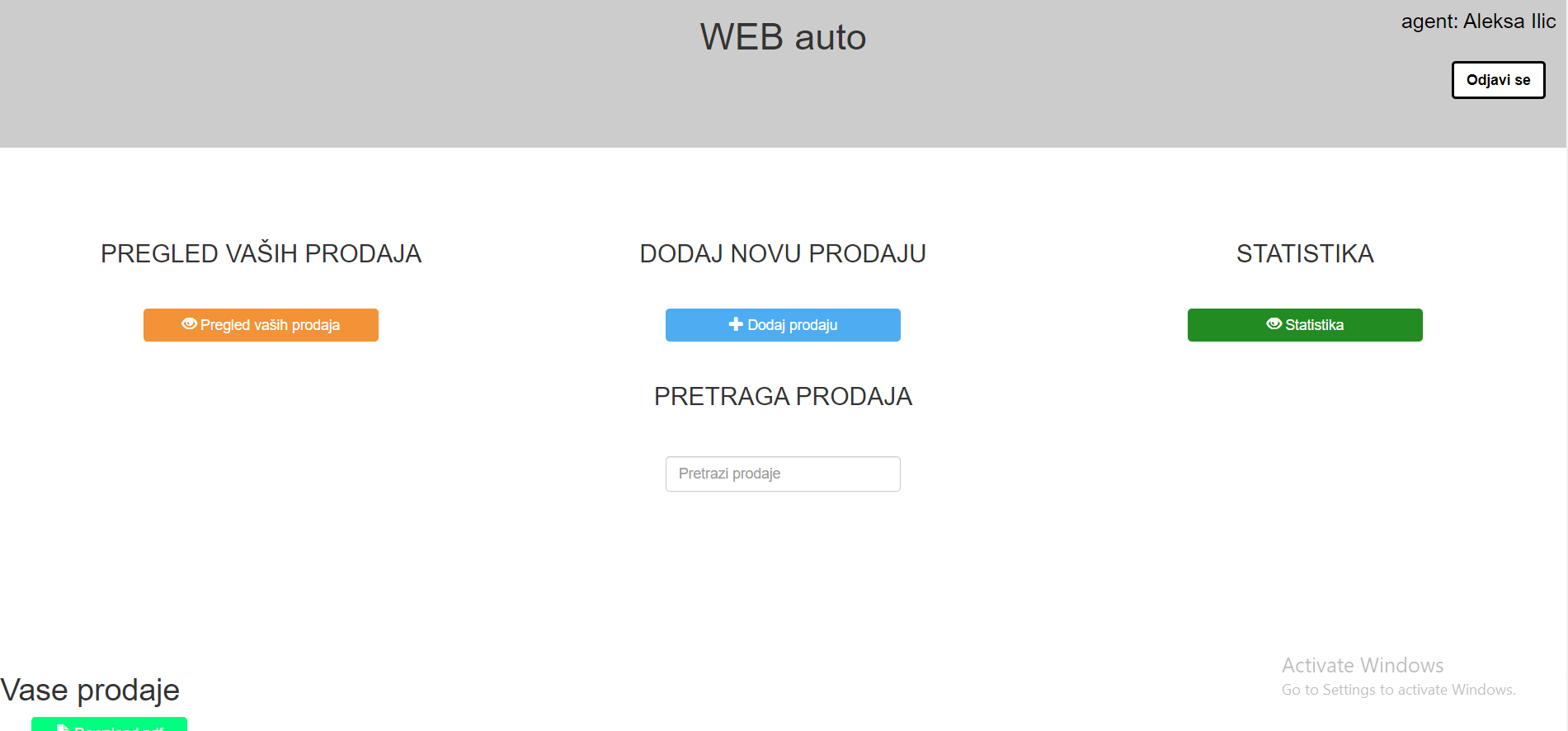


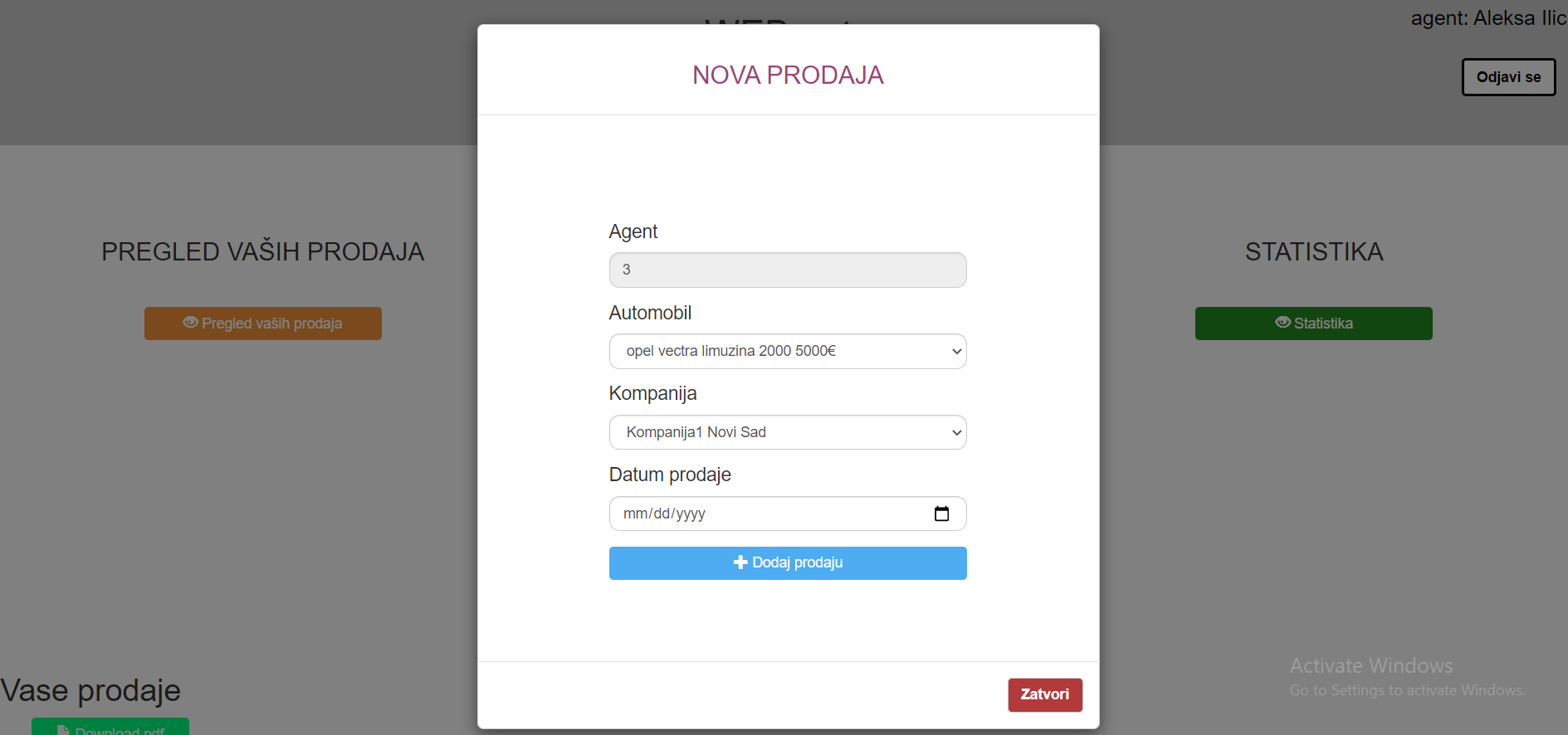
Na dnu stranice, prikazana je statistika prodaja svih agenata, broj prodaja po mesecu, prodaje po tipu automobila i iznos prodaje po kompanijama.





Agentima je omogućeno da unose nove prodaje klikom na dugme dodaj prodaju.





Na dnu stranice takođe postoji statistika ali sa podacima koji zavise od ulogovanom agenta.

I administratoru i agentu je omogućeno da, klikom na dugme „download pdf“, u PDF-u preuzmu tabelu “PRODAJE”.