UNIVERZITET U BEOGRADU

**FAKULTET ORGANIZACIONIH NAUKA**

**Internet tehnologije**

Tema: eUgostiteljstvo – Milanovic BNB

**Mentor: Student:**

Beograd, 2020.

Sadržaj

[1 Opis aplikacije - 4 -](#_Toc32140282)

[2 Opis tehnologija korišćenih u aplikaciji - 5 -](#_Toc32140283)

[2.1 Bootstrap - 5 -](#_Toc32140284)

[3 HTML - 8 -](#_Toc32140285)

[3.1 CSS - 8 -](#_Toc32140286)

[3.2 JAVASCRIPT - 9 -](#_Toc32140287)

[3.3 PHP - 10 -](#_Toc32140288)

[3.4 AJAX - 13 -](#_Toc32140289)

[3.5 SQL - 14 -](#_Toc32140290)

[4 Korisnicko uputstvo - 15 -](#_Toc32140291)

[4.1 DataTables - 17 -](#_Toc32140292)

[4.2 Google Chart - 19 -](#_Toc32140293)

[4.3 Google Maps - 21 -](#_Toc32140294)

[5 Baza podataka - 22 -](#_Toc32140295)

[6 Relacioni model - 23 -](#_Toc32140296)

[7 Veb servis 24](#_Toc32140297)

[8 Slučajevi korišćenja 30](#_Toc32140298)

[8.1.1 SK1 – Logovanje 30](#_Toc32140299)

[8.1.2 SK2 – Pregled apartmana u ponudi 30](#_Toc32140300)

[8.1.3 SK3 – Brisanje apartmana 30](#_Toc32140301)

[8.1.4 SK4 – Dodavanje apartmana 31](#_Toc32140302)

[8.1.5 SK5 – Pretraživanje apartmana 31](#_Toc32140303)

[8.1.6 SK6 – Kupovina apartmana 31](#_Toc32140304)

[8.1.7 SK7 – Pretraživanje svojih Rezervacija 31](#_Toc32140305)

[8.1.8 SK8 – Pretraživanje svih Rezervacija 32](#_Toc32140306)

[8.1.9 SK9 – Brisanje porudzbine 32](#_Toc32140307)

[8.1.10 SK10 – Otkazivanje porudzbine 32](#_Toc32140308)

[8.1.11 SK11 – Izmena podataka apartman 33](#_Toc32140309)

[8.1.12 SK13 – Pretraga Gosta 33](#_Toc32140310)

[8.1.13 SK14 – Pregled detalja apartman 33](#_Toc32140311)

[8.1.14 SK16 – Logout 34](#_Toc32140312)

[8.2 Dijagram sekvenci 35](#_Toc32140313)

[8.2.1 DS1 – Logovanje 35](#_Toc32140314)

[8.2.2 DS2 – Pregled liste apartmana 35](#_Toc32140315)

[8.2.3 DS3 – Brisanje apartmana 36](#_Toc32140316)

[8.2.4 DS4 – Dodavanje apartmana 37](#_Toc32140317)

[8.2.5 DS5 – Pretraživanje apartmana 37](#_Toc32140318)

[8.2.6 DS6 – Pretraživanje Gosta 38](#_Toc32140319)

[8.2.7 DS8 – Pretraživanje Rezervacija 39](#_Toc32140320)

[8.2.8 DS9 – Brisanje Gosta 40](#_Toc32140321)

[8.2.9 DS10 – Izmena podataka apartman 41](#_Toc32140322)

[8.2.10 DS11 – Pregled liste Rezervacija 42](#_Toc32140323)

[8.2.11 SK12 – Logout 43](#_Toc32140324)

# Opis aplikacije

Aplikacija MilanovicBNB omogucava korisnicima da vide sve knjige u ponudi, sortirane po tipu, I detalje o zeljenoj knjizi. Zeljene knjige korisnik moze dodati u korpu, poruciti I pratiti status svojih Rezervacija.

Za porucivanje je neophodno registrovati se I logovati.

Postoje tri tipa korisnika aplikacije:

* Neregistrovani posetioci
* Kupci
* Administrator

Aplikacija je pisana po MVC paternu kako bi se olakšalo pisanje i održavanje koda aplikacije. Poseduje interne web servise, kao i AJAX zahteve koji pozivaju sopstvene i eksterne web servise. U folderu controllers smešteni su kontrolori aplikacije. Logika aplikacije smeštena je u kontrolorima koji predstavljaju servise aplikacije koje Flight učitava i u njima su mapirane rute HTTP poziva koje klijenti mogu slati aplikaciji. U folderu models nalaze se modeli podataka sa kojima aplikacija radi.

# 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 budeo 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 grad. 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 mišeljenje 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 grada, 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.

# HTML

HTML, HyperText Markup Language, je standarizovani jezik koji se koristi pri strukturiranju tekstova, medija i ugrađenih objekata u web stranice i elektronsku poštu. Kao modifikovanu i pojednostavljenu verziju SGML jezika, HTML standarizuje i održava World Wide Web Consortium (W3C). Originalnu verziju HTML-a kreirao je Tim Berners-Lee, a prva zvanična verzija je izašla juna 1993 godine.

HTMl jezik je sačinjen od običnog teksta i tagova. Iako se HTML znakovi (tags) često nazivaju kodom, tehnički HTML nije kod jer kompjuterski kod su instrukcije koje od računara traže da izvrši određenu operaciju. Svrha HTML znakova je da se se struktura dokumenta "označi" tako da bi korisnikov agent (user agent), tj. internet pretraživač mogao da prepozna strukturu dokumenta i ispravno je prikaže u prozoru internet pretraževača kojeg korisnik koristi.

Za dodatno modifikovanje web stranica, do sada su izašli neki dodaci:

* CSS ili Cascading Style Sheets za izgled i položaj prezentacije
* Skriptni jezici (Javascript, VBScript) za omogućavanje dinamičnosti i interaktivnosti na web prezentacijama,
* DOM ili Document Object Model, koji označava vezu između skripte i elementa na stranici
* Gornji dodaci, u paketu sa HTML jezikom, ponekad se nazivaju DHTML ili Dinamični HTML.

## CSS

CSS je u osnovi skup instrukcija koje definišu kako neki HTML dokument treba da se prikaže. Najbolja stvar kod CSS-a je da se on može deflnisati u spoljašnjoj datoteci, ili listu (sheet). Pretraživač onda može da pročita ovaj spoljašnji list i da primeni sadržaj na određeni sadržaj datoteke. Na taj način se stil i formatiranje strane odvajaju od samog sadržaja. Ovo je vrlo bitan koncept kod dizajna web strana. Ovo Vam pruža moćan i detaljan način da ažurirate svoj web sajt. U datoteci sa stilovima možete definisati da se za tekst primenjuje font Arial, da je veličina slova 12 piksela, da su slova crvene boje, da su masna, da je tekst levo poravnat i da je pozadina plava. Ako ovu datoteku pridružite svim stranama na svom sajtu, tekst će se uvek prikazivati na taj način. Ako promenite mišljenje, možete da promenite jednu datoteku i da zadate da font bude Times, da veličina slova bude 28 piksela, da su zelene boje, pisana kurzovim i desno poravnata. Upamtite tu novu datoteku i ceo web sajt se automatski ažurira. CSS takođe omogućava da radite neke stvari koje u HTML-u nisu moguće. Za svoju stranu možete definisati nivo kroz koji ćete se kretati klizačem. Na taj način možete da imate prozor u sredini strane sa sadržajem kroz koji se možete kretati uz pomoć klizača. Za svaku ivicu u okviru tabele možete definisati različitu boju. Možete podesiti da se kod oznaka za nabrajanje koristi korisnički definisana grafika, ili možete obrisati podvlačenje hiperlinkova. Premda se sa svakom novom verzijom pretraživača poboljšava podrška koju oni imaju za CSS, ipak neće svi pretraživači prikazati sve elemente CSS-a, niti će sve elemente prikazati na isti način. Uvek je dobro da svoju stranu testirate u različitim pretraživačima. Jedna od bitnih ideja u vezi CSS-a je da ako pretraživač ne podržava neki stil, informacije su i dalje tu, ali se ne koriste stilovi koji nedostaju. IE 5 i Netscape 6 i novije verzije uglavnom prikazuju većinu CSS elemenata. Izvesna ograničenja postoje u verziji Netscape 4. CSS Vam daje neverovatnu kontrolu nad stranama. W3C sada preporučuje da za formatiranje strana koristite CSS. Trebalo bi da pažljivo razmotrite mogućnosti primene CSS-a kod dizajna.

## JAVASCRIPT

Java Script pripada grupi jezika za skriptovanje, pre svega klijentske strane, mada se može izvršavati i na serveru (runat=“server“). 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.

JavaScript dodaje interaktivnost veb prezentaciji na strani klijenta. Na primer, korišćenjem JavaScripta je moguće odgovarati na akcije korisnika u samom veb čitaču. Bitno je naglasiti da JavaScript program može da se izvrašava samo u okviru veb čitača i nigde drugde. Pri kreiranju JavaScript koda može se koristiti bilo koji ASCII editor.

Polazni naziv JavaScript jezika bilo je LiveScript, ali u saradnji Netscape-a i Sun-a, ime je izmenjeno u JavaScript, zbog namere da Java i JavaScript budu jezici Interneta i internet programiranja.

JavaScript programi se koriste da bi detektovali i reagovali na događaje koji su inicirani od strane korisnika, kao npr. prelazak mišem preko linka i grafičkog objekta na strani. JavaScript kod može da poboljša kvalitet sajta navigacionim pomagalima, skrolovanim porukama, dialog prozorima, dinamičnim slikama, koliicima za kupovinu (shopping cart), itd. JavaScript omogućava korisnicima da kontrolišu izled veb strane dok se dokument „tumači“ od strane čitača. Uz pomoć JavaScripta, možemo da proverimo tačnost podataka koje je korisnik ukucao u formu, pre nego što se ti podaci prebace na server. JavaScript testira brauzaer korisnika da vidi da li ima neophodne dodatke (plug inns) za učitavanje tražene strane i ako ih nema, JavaScript usmerava korisnika na drugi sajt da ih nabavi. JavaScript služi kao programski jezik. Njegovo jezgrosadrži osnovne elemente programskih jezika kao što su promenljive, tipovi podataka, kontrolne petlje, if/else izjave, switch izjave, funkcije i objekte. JavaScript se koristi za aritmetičke operacije, manipulacije datumom i vremenom, nizovima, stringovima i objektima. On takođe čita i upisuje vrednosti kuki fajlova i dinamički kreira HTML na osnovu vrednosti kukija.

## PHP

Server - side skripting je web server tehnologija koja omogućava da se korisnički zahtevi obrađuju pomoću skripti koje se izvršavaju na serverskoj strani kako bi se generisale dinamičke stranice. Najčešće se koristi da bi se interaktivne web stranice povezale sa bazama podataka, radi identifikacije korisnika, ažuriranja sadržaja, stvaranja raznih diskusionih grupa i još zbog mnogo toga. Razlika između skriptovanja na serverskoj i korisničkoj strani je u tome što se kod korisničke strane skripte izvršavaju u web browser-u korisnika, najčešće pomoću JavaScript-a.

Najzastupljenije tehnike serverske strane:

* Common Gateway Interface
* Active Server Pages
* Java Servlet i Java Server Pages
* PHP

PHP je open source jezik koji se koristi za razvoj server-side aplikacija, kao i dinamičkog Web sadržaja. PHP dozvoljava interakciju sa velikim brojem relacionih baza podataka kao što su MySQL, Oracle, IBM D2, Microsoft SQL Server, PostgreSQL i SQLite. PHP radi na većini operativnih sistema današnjice, kao što su UNIX, Linux, Windows i Mac OS i može da interaguje sa većinom Web servera.

Personal Home Page Tools je predstavio PHP 1995. godine, a dve godine kasnije su razvoj nastavila dva Izraelska programera. Od 1999. godine se zasniva na Zend engine-u. PHP je prvo bio akronim za Personal Home Page Tools, da bi kasnije promenio naziv u Hypertext Preprocessor.

PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. Nalikuje jeziku C++ u sclanlu da dozvoljava i čisto-proceduralno programiranje ali omogućava i korišćenje klasa i drugih koncepata objektno orijentisanog programiranja (nasleđivanje, apstraktne metode, interfejsi itd.).

Danas, PHP je instaliran na više od 20 miliona sajtova i preko million web servera. Poslednja realizovana stabilna verzija je 5.2.6 iz maja 2008.god.

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:

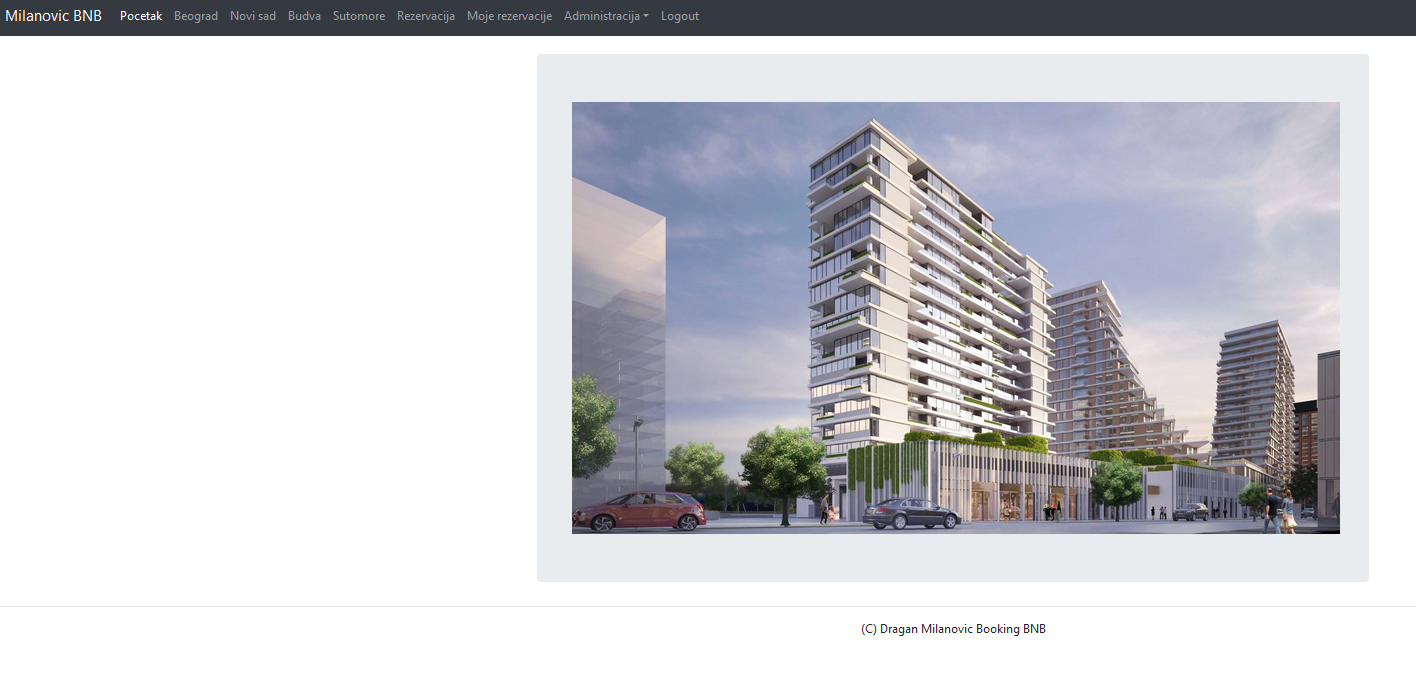
* Validacija podatka u realnom vremenu - Podaci na formama kao što su: user IDs, serial numbers, postal codes, i ostali koji zahtevaju validaciju od strane servera, mogu biti provereni pre submit- ovanja forme.
* Autocompletion i autosuggest - Određeni delovi podataka koji se unose od strane korisnika(posetioca): 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

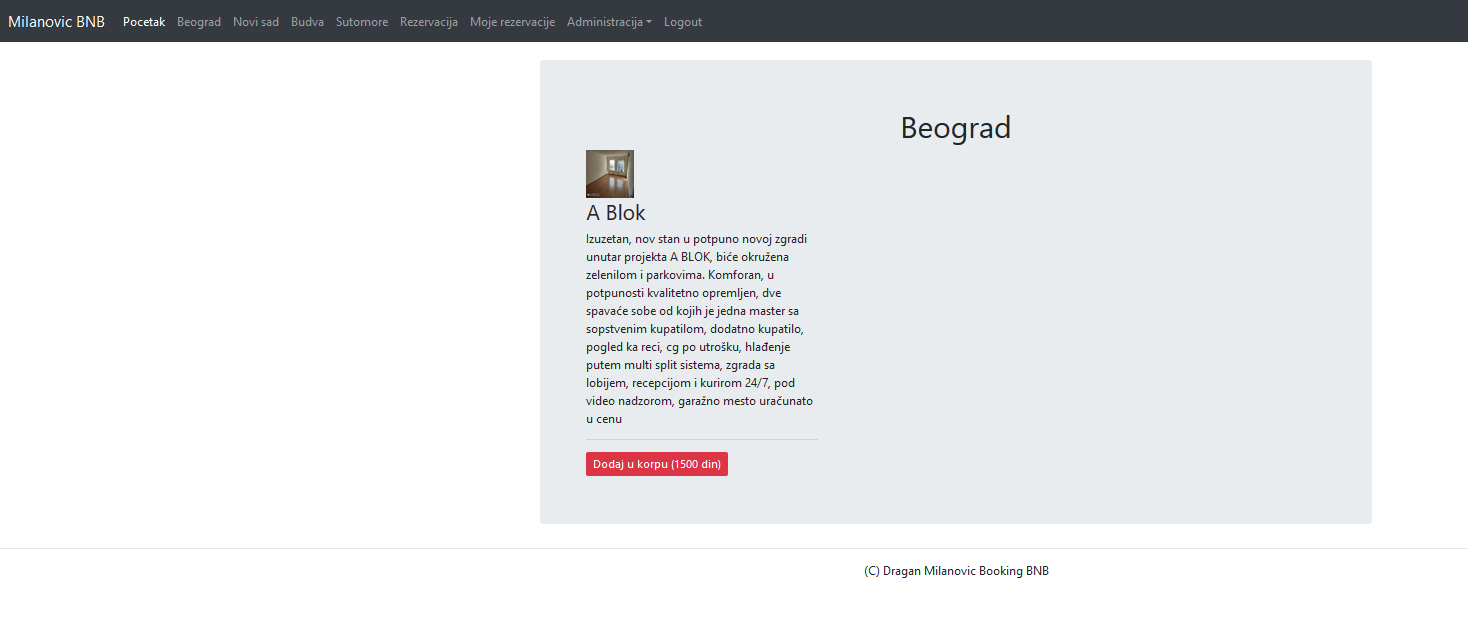
SQL je standardni programski jezik za pristup i manipulaciju podacima iz relacionih baza podataka i podržan je od strane svih servera relacionih baza podataka. U relacionim bazama podaci su skladišteni preko niza tabela. Svaka tabela sadrži kolone koje opsiuju atribute podataka, a svaki red je instanca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

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

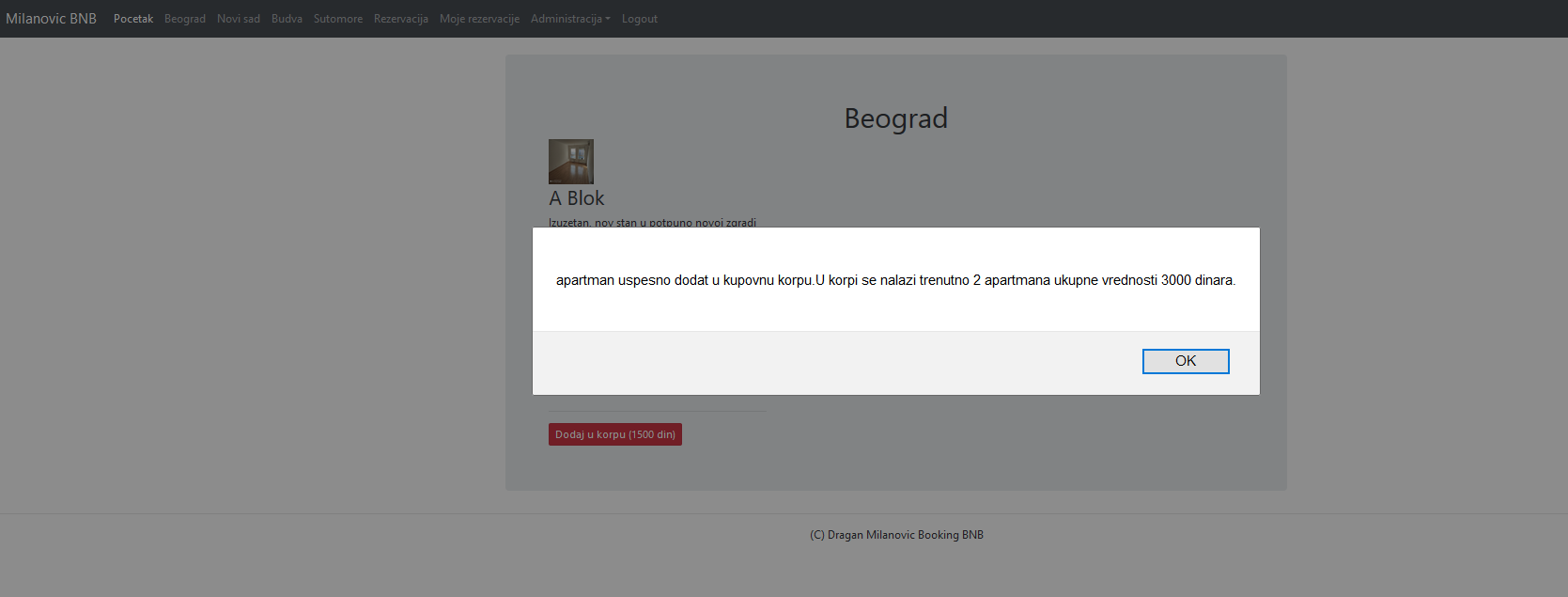
# Korisnicko uputstvo



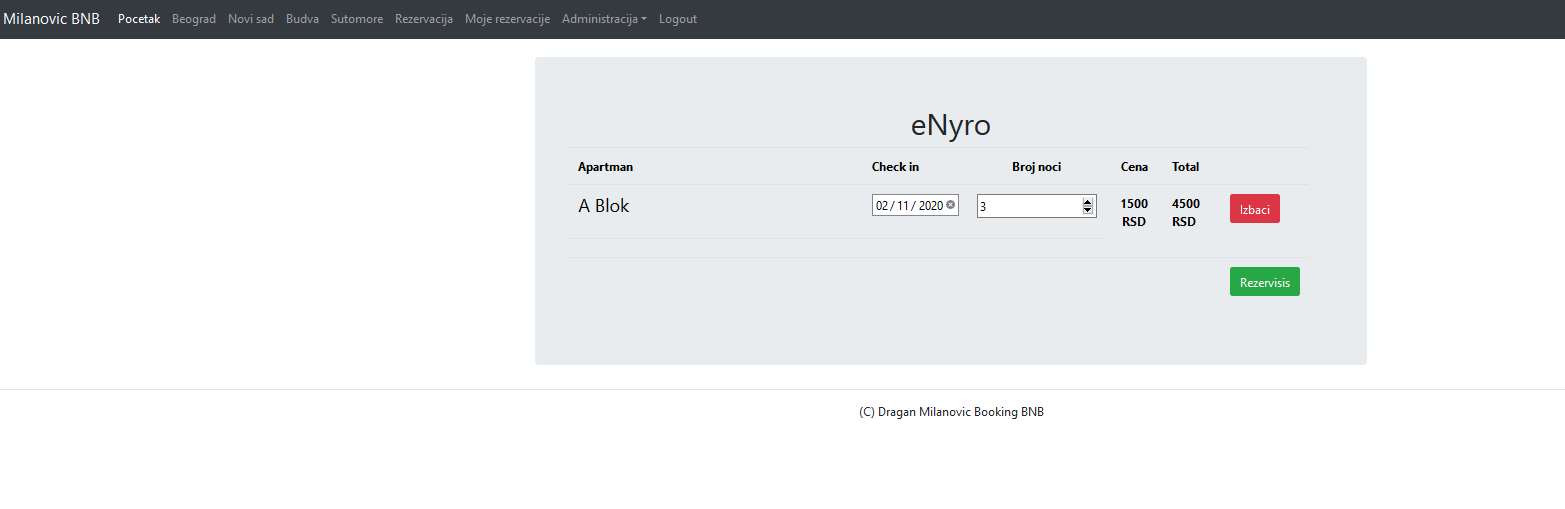
Pocetna strana sajta



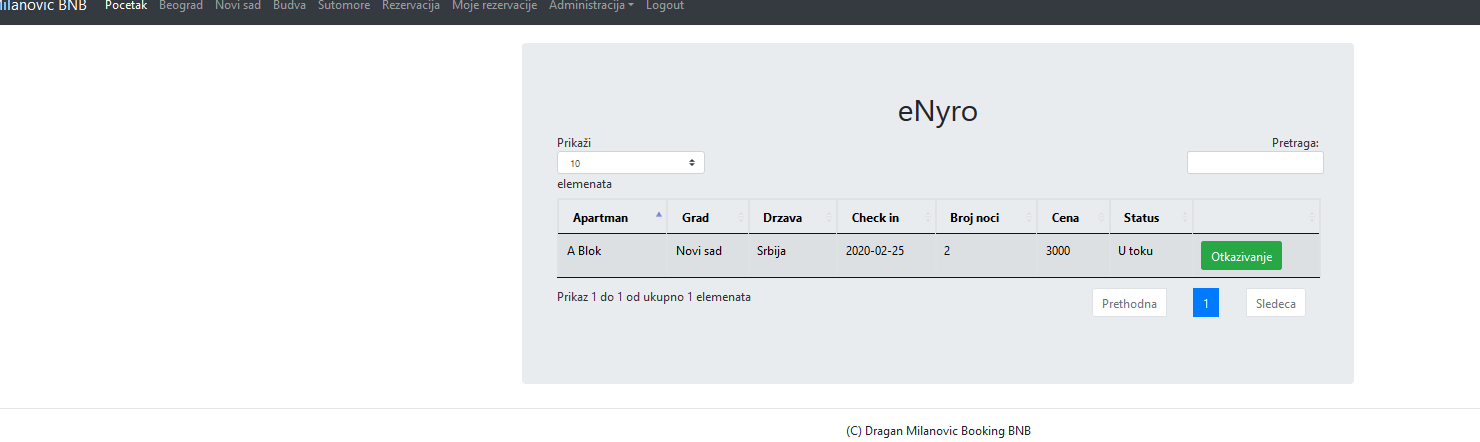
Apartmani u Beogradu



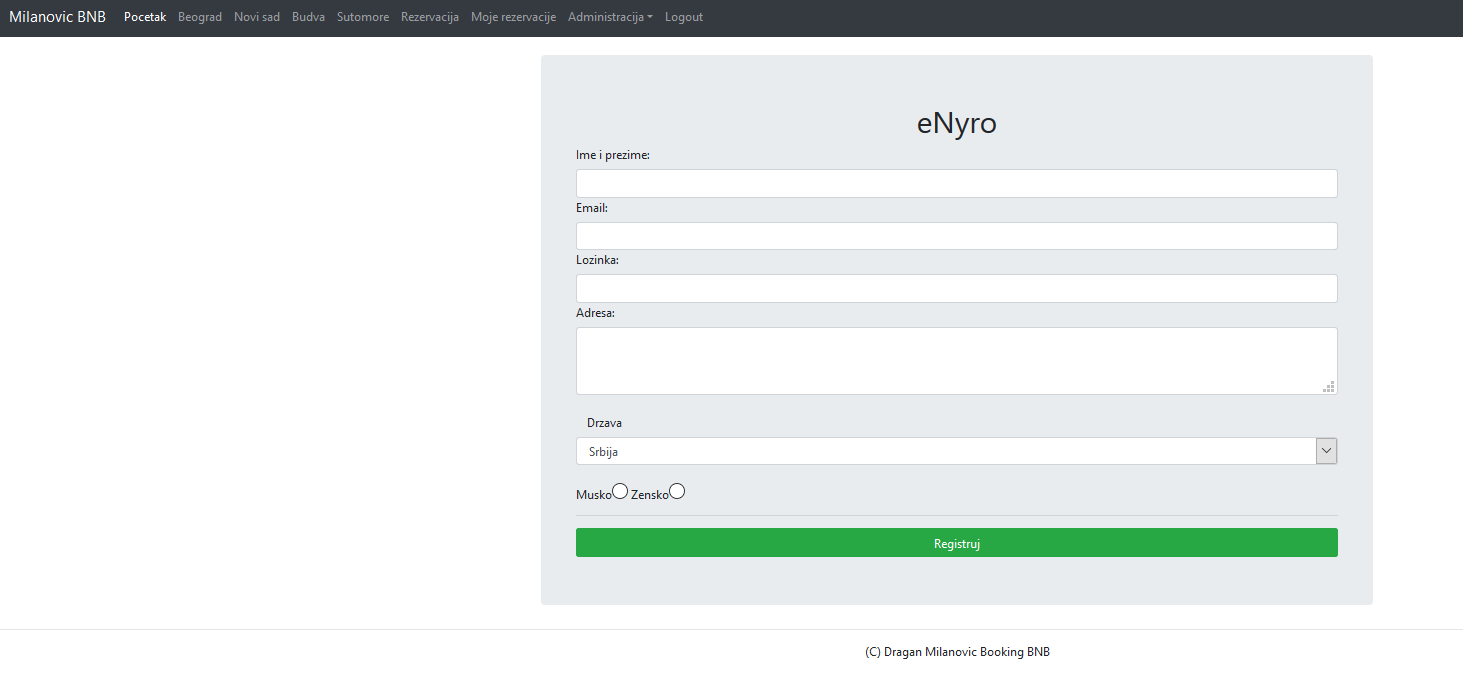
Stavljanje zeljenog apartmana u kupovnu korpu



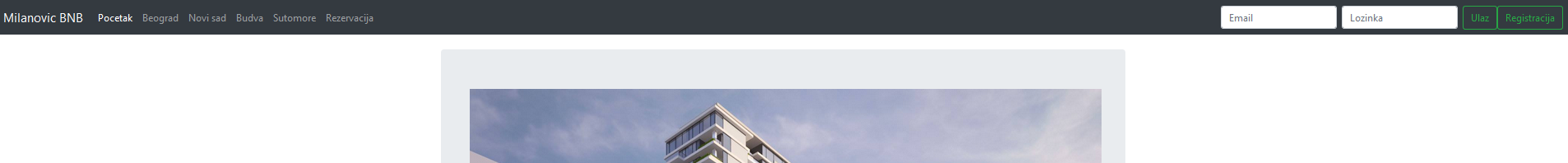
Rezervisanje



Pregled Rezervacija



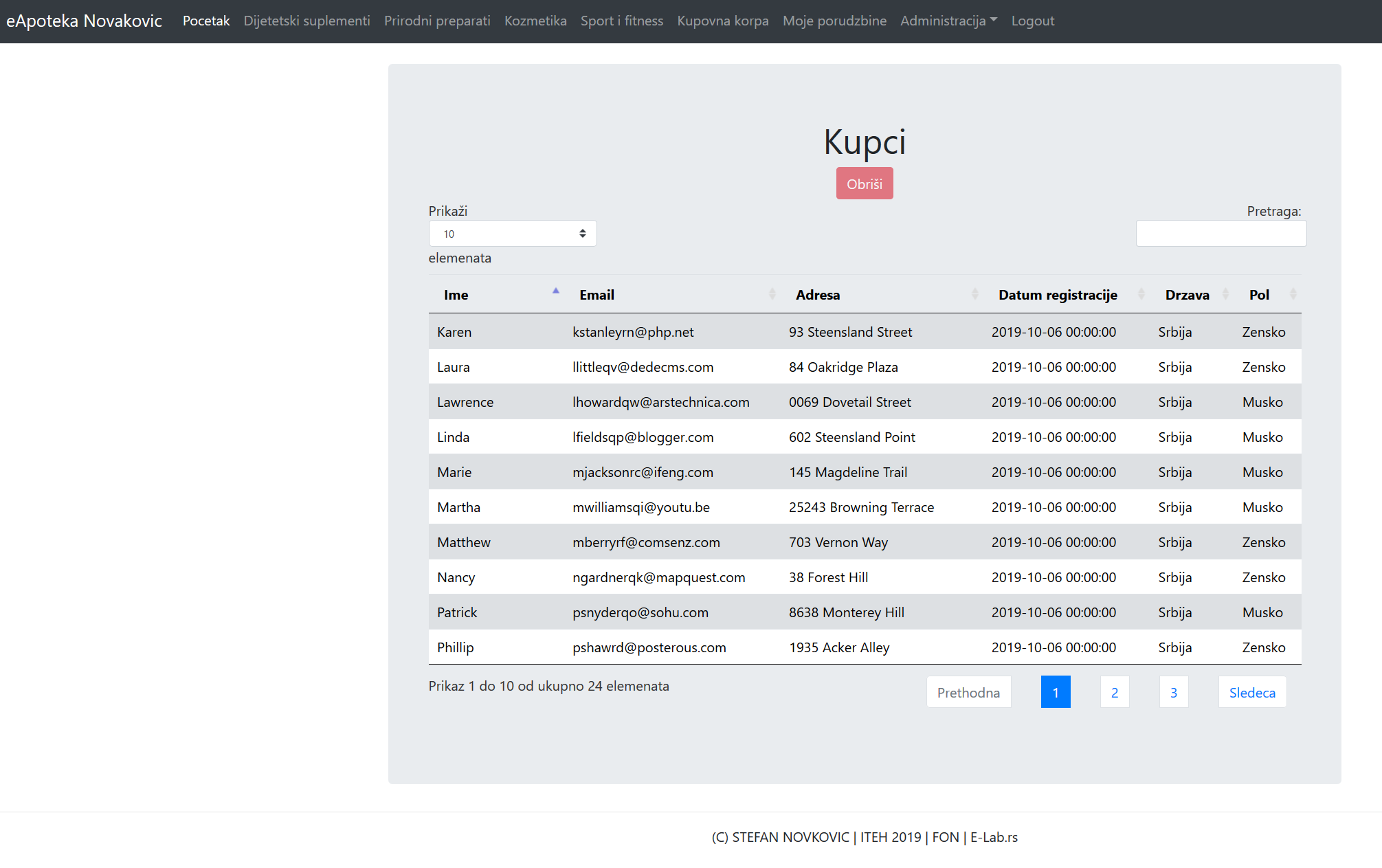
Registracija



Logovanje

## DataTables

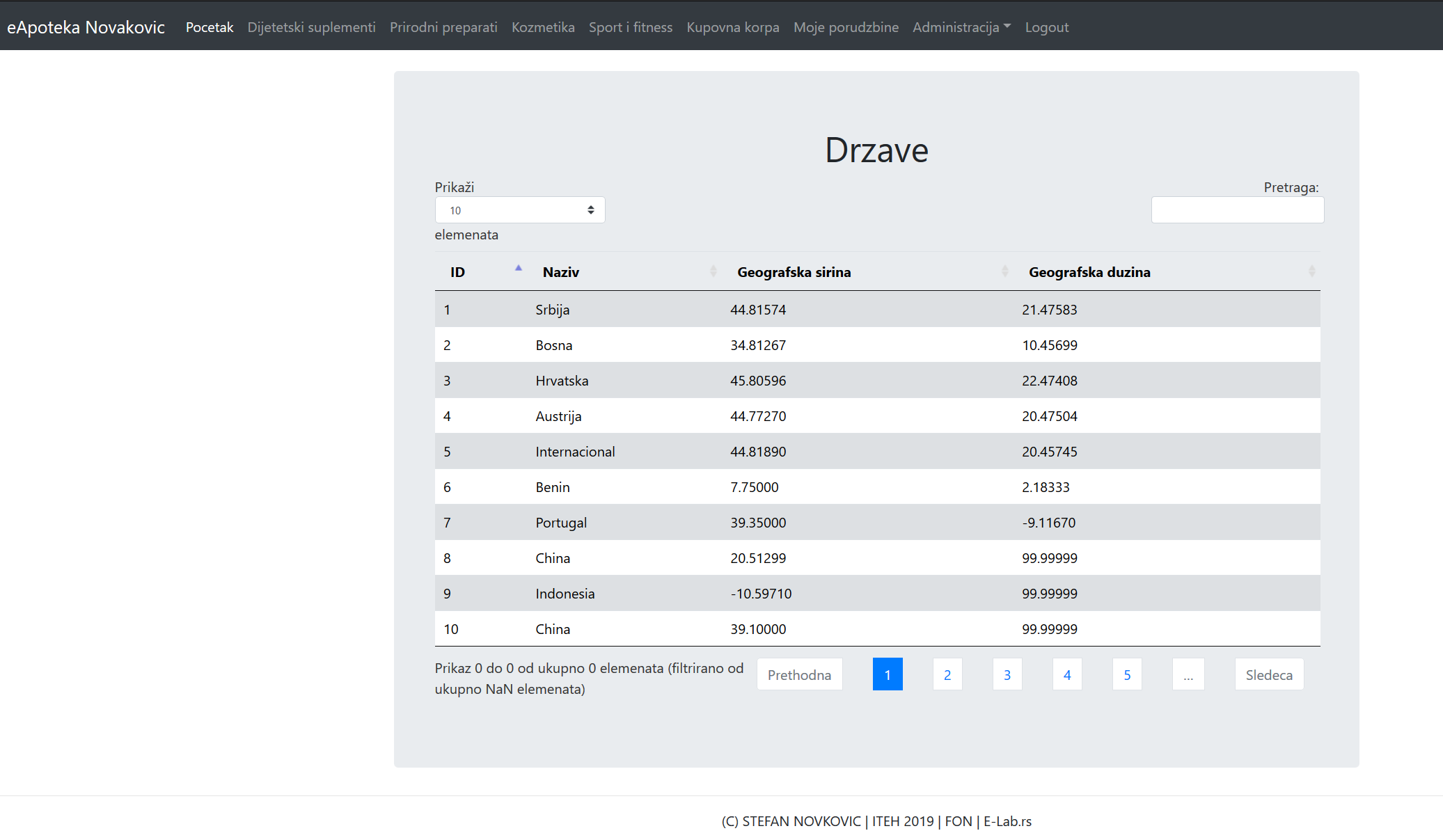
Na stranici Administracija->Kupci implementirana je DataTables tabela u kojoj se prikazuju podaci iz tabela Kupci i Drzava.



U bazi ima oko 50 unosa, a korisnik sam bira koliko ce mu biti prikazano po u tabeli po stranici. Implementiran i jEditable plugin koji omogucava CRUD operacije. Podaci se dobijaju iz dve tabele – kupci I drzave, pomocu JOIN operacije. Implementirana je jezicka lokalizaija tabele, i stilizacija pomocu Bootstrap-a.

Druga tabela je implementirana na strain Administracija->drzave. U tabeli su prikazani podaci o drzavama dobijeni u JSON formatu pomocu AJAX poziva. Nad tabelom je implementirana scrolable funkcionalnost koja omogucava dobijanje podataka kako korisnik skroluje tabelu, cime su izbegnuti dodatni klikovi kao kod klasicne paginacije.

Pretraga i sortiranje tabele su takodje Server Side, pa se prilikom svake o tih aktivnosti pravi novi AJAX upit.



U tabeli su prikazani podaci o drzavama dobijeni u JSOn formatu nakon AJAX poziva. Nad tabelom je implementirana scrolable funkcionalnost koja omogucava dobijanje podataka kako korisnik skroluje tabelu, cime su izbegnuti dodatni klikovi kao kod klasicne paginacije. Pretraga i sortiranje tabele su takodje Server Side, pa se prilikom svake o tih aktivnosti pravi novi AJAX upit.

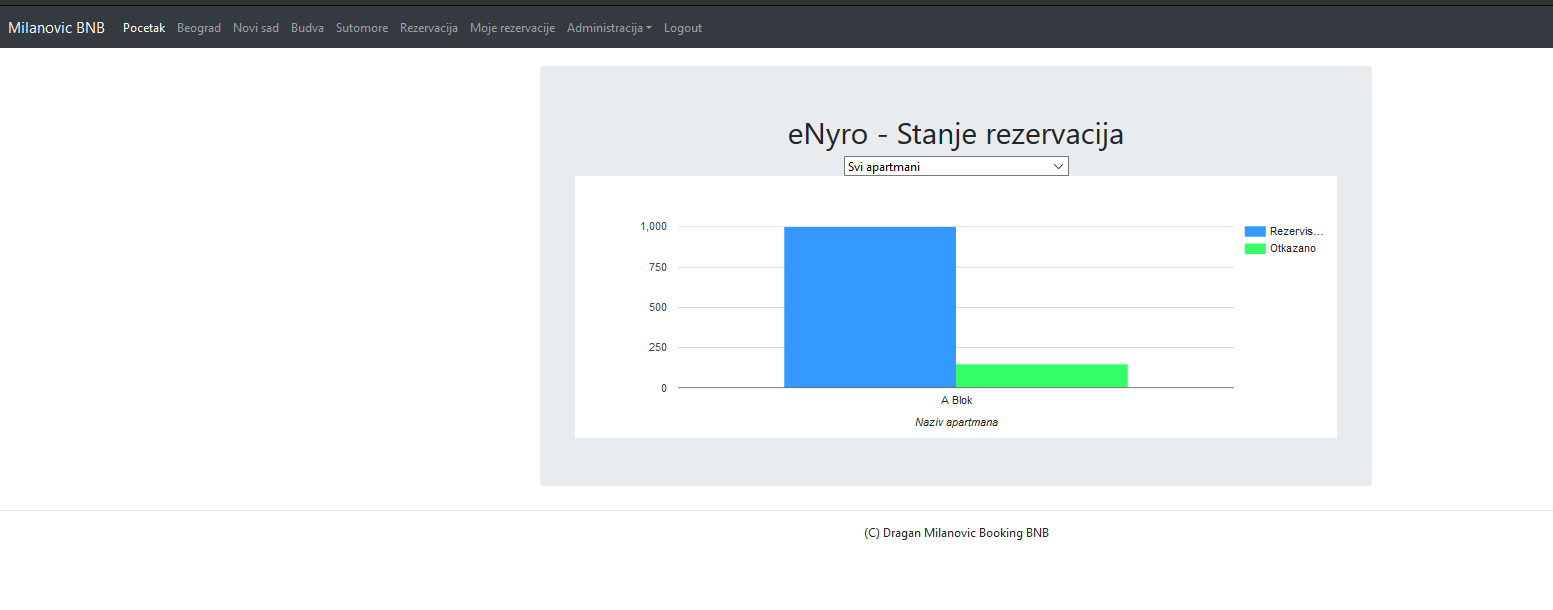
## Google Chart

Odabirom opcije Stanje rezervacija prikazuje se koliko za svaki apartman iz ponude postoji rezervacija. Takodje za svaki apartman postoji je prikazano I koliko je do sada otkazano.

Podaci za grafik se dobijaju pozivom lokalnog veb servisa http://enyro.localhost.com/API/statistika\_apartmani a sam grafik pozivom Google Chart veb servisa. Svi pozivi se obavljaju asihrono:



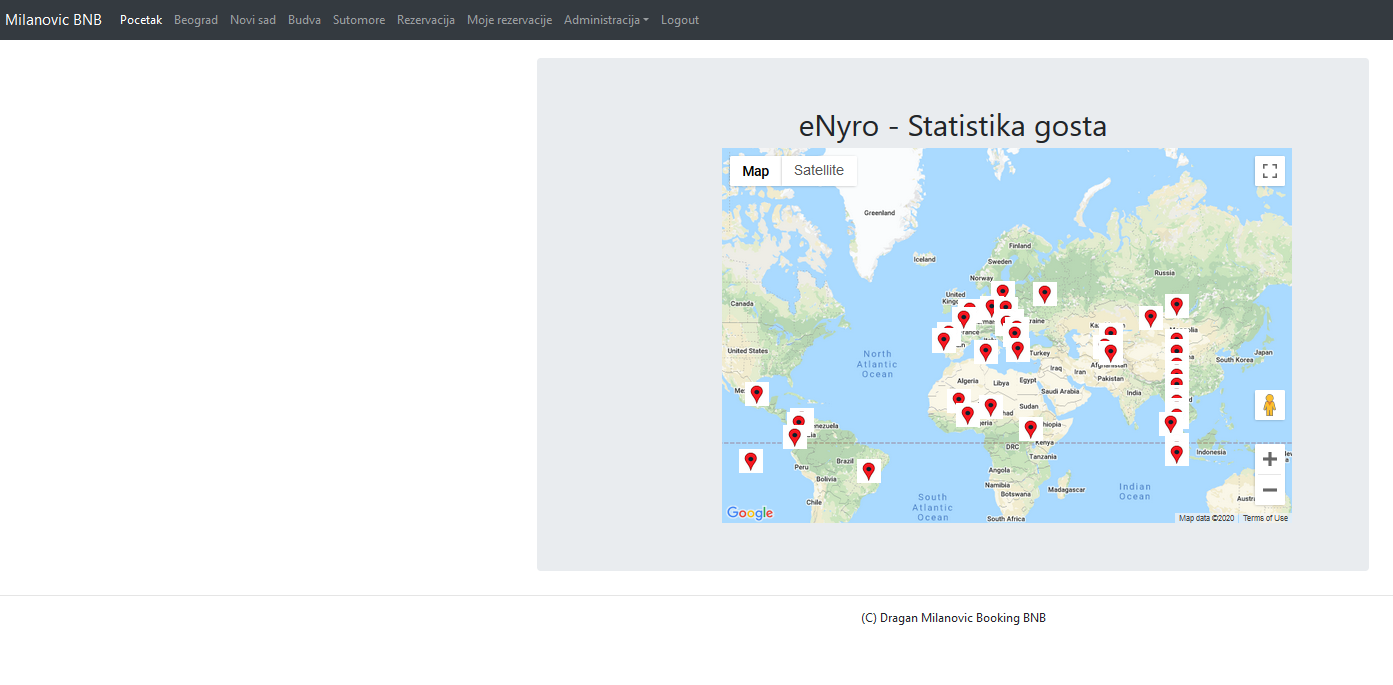
Podatke u grafiku je moguce promeniti odabirom zeljene opcije u padajucem meniju:



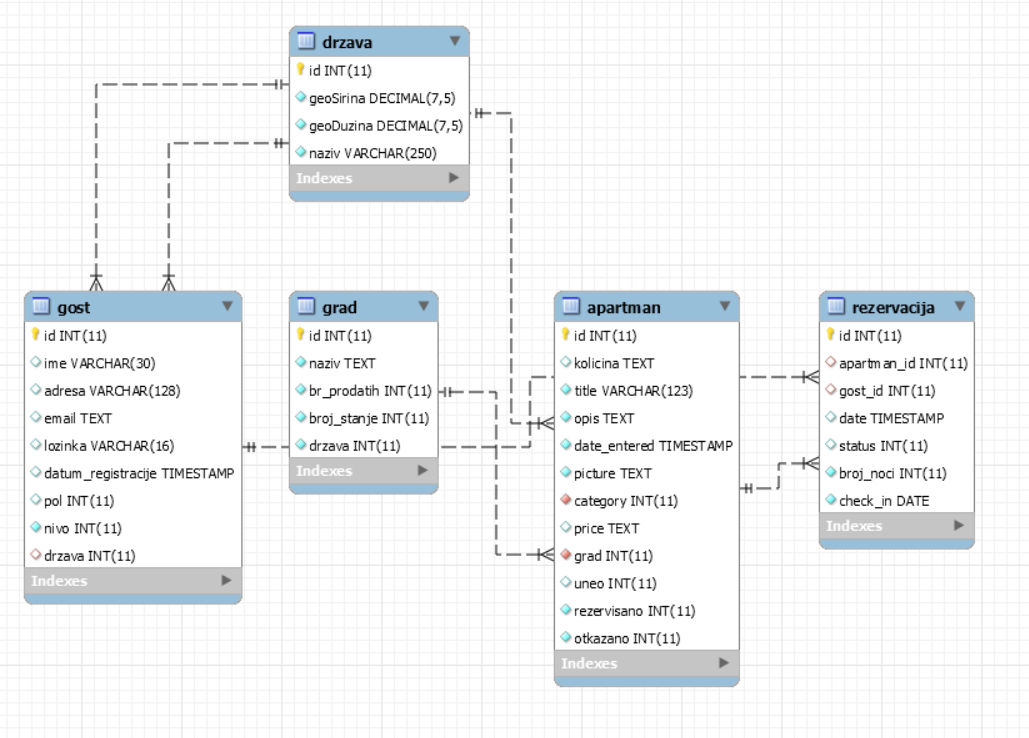
Prilikom promene u padajucem meniju pravi se nov AJAX poziv i grafik se osvezava bez osvesavanja cele stranice.

## Google Maps

Za svaki apartman se prilikom unosa bira drzava odakle potice, a na stranici Statistika drzava prikazana je Google Mapa sa osnacenim lokacijama odakle obrok potice. Podaci se dobijaju AJAX pozivom lokalnog veb servisa:



# Baza podataka



# Relacioni model

**Gost** (id, ime, adresa, email, lozinka, datum\_registracije, pol, nivo, drzava)

**Apartman** (id, naziv, opis, datum\_unosa, grad, slika, kolicina, raposlozivo, cena, prodato)

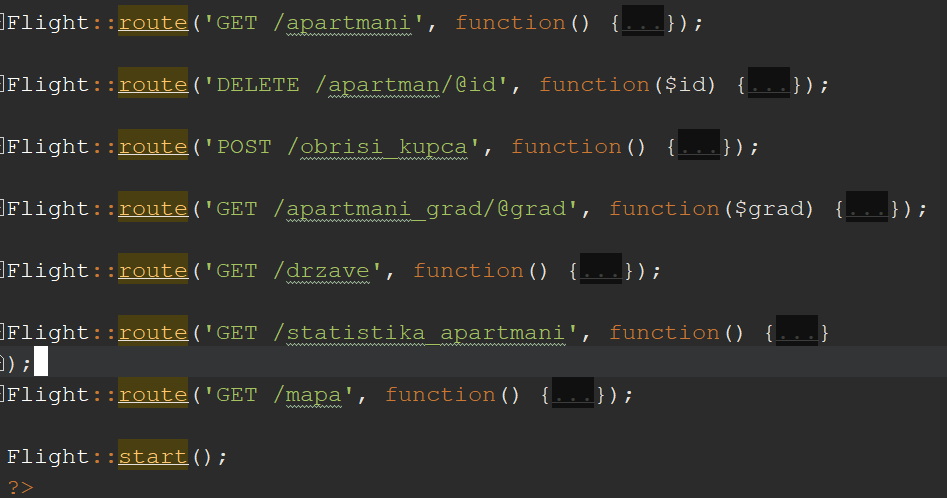
**Drzava** (id, naziv, geoSirina, geoDuzina)

**Rezervacija** (id, apartman\_id, Gost\_id, datum\_porucivanja)

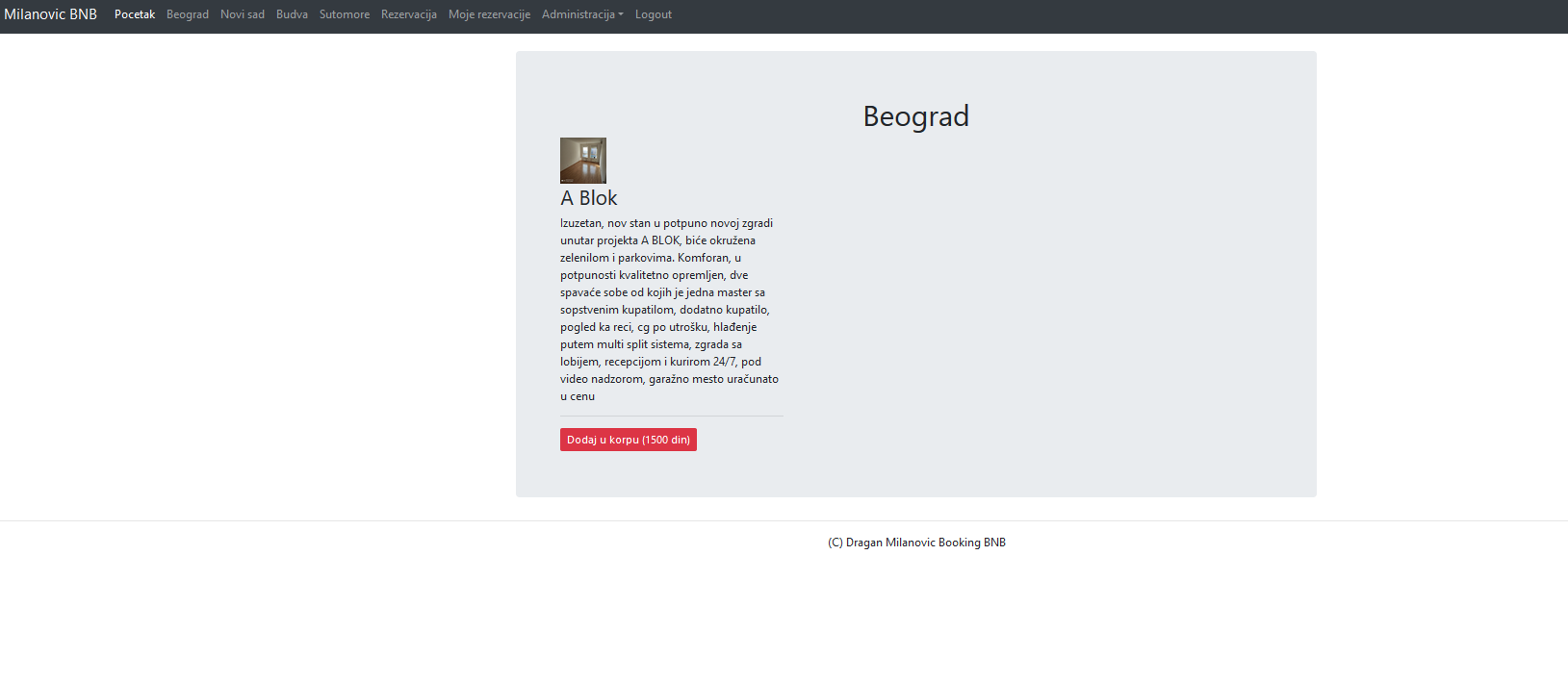
**Grad** (id, naziv)

# Veb servis

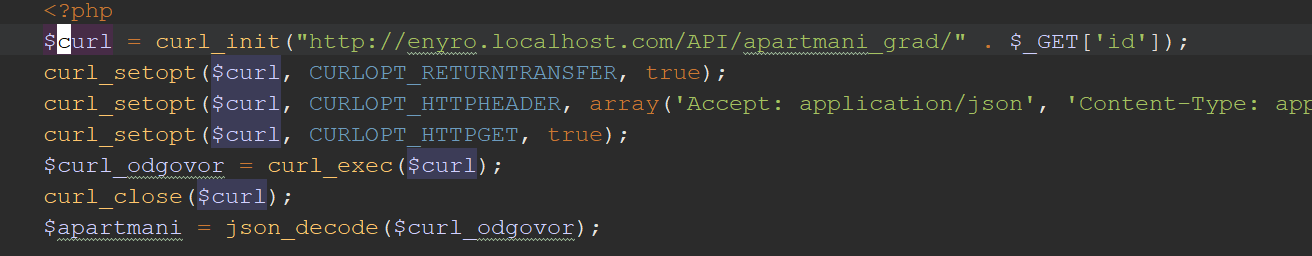
Veb servis se sastoji od sledecih ruta:



Baza veb servisa se sastoji od 4 medjusobno povezane tabe spoljnim kljucevima. Veb servis podatke vraca u JSON formatu. Klijent poziva sopstveni veb servis na stranicama sa apartmanima u svim gradovima, dobijeni JSON parsira i prikaze na sledeci nacin:



Veb srvis se poziva pomocu CURL-a na sledeci nacin:



Sopstvei veb servis se poziva i AJAX-om na stranici za Administraciju apartmana kada se klikne na DELETE .

U fajlu index.php nalaze sve sve funkcije WS-a, u okviru ruta. Za rutiranje koriscene su GET, DELETE i POST http metode.

U okviru svake f-je WS-a se salje HTTP header: header("Content-Type: application/json; charset=utf-8");

|  |  |
| --- | --- |
| Opis funkcije | Pregled svih apartmana |
| HTTP metoda | GET |
| URL | http://MilanovicBNB.com/API/apartmani |
| URL parametri | nema |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima podatke o apartmanu:  Primer:  [ { "apartman\_id":"54", "id":"1", "kolicina":"Dvosoban 90m2", "title":"A Blok", "opis":"Izuzetan, nov stan u potpuno novoj zgradi unutar projekta A BLOK, bi\u0107e okru\u017eena zelenilom i parkovima. Komforan, u potpunosti kvalitetno opremljen, dve spava\u0107e sobe od kojih je jedna master sa sopstvenim kupatilom, dodatno kupatilo, pogled ka reci, cg po utro\u0161ku, hla\u0111enje putem multi split sistema, zgrada sa lobijem, recepcijom i kurirom 24\/7, pod video nadzorom, gara\u017eno mesto ura\u010dunato u cenu", "date\_entered":"2020-02-09 02:07:15", "picture":"slike\/apartmani\/original\/948134865a-blok-savada-nov-lux-dvosobni-stan-5425635330592-71790198976.jpg", "category":"0", "price":"1500", "grad":"1", "uneo":null, "rezervisano":"1000", "otkazano":"150", "naziv":"Novi sad", "br\_prodatih":"15", "broj\_stanje":"29", "drzava":"1" } ] |
| Format izlaznih parametara | application/json |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Opis funkcije | Pregled apartmana iz nekog grada |
| HTTP metoda | GET |
| URL | http://MilanovicBNB.com/API/apartmani\_grad/@id\_grada |
| URL parametri | Int $id\_grada |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima podatke o apartmanu:  Primer:  [ { "apartman\_id":"54", "id":"1", "kolicina":"Dvosoban 90m2", "title":"A Blok", "opis":"Izuzetan, nov stan u potpuno novoj zgradi unutar projekta A BLOK, bi\u0107e okru\u017eena zelenilom i parkovima. Komforan, u potpunosti kvalitetno opremljen, dve spava\u0107e sobe od kojih je jedna master sa sopstvenim kupatilom, dodatno kupatilo, pogled ka reci, cg po utro\u0161ku, hla\u0111enje putem multi split sistema, zgrada sa lobijem, recepcijom i kurirom 24\/7, pod video nadzorom, gara\u017eno mesto ura\u010dunato u cenu", "date\_entered":"2020-02-09 02:07:15", "picture":"slike\/apartmani\/original\/948134865a-blok-savada-nov-lux-dvosobni-stan-5425635330592-71790198976.jpg", "category":"0", "price":"1500", "grad":"1", "uneo":null, "rezervisano":"1000", "otkazano":"150", "naziv":"Novi sad", "br\_prodatih":"15", "broj\_stanje":"29", "drzava":"1" } ] |
| Format izlaznih parametara | application/json |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje apartmana |
| HTTP metoda | DELETE |
| URL | http://MilanovicBNB.com/API/apartman/@id\_apartmana |
| URL parametri | INT id\_apartmana npr 39 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Poruka o uspesnom/neuspesnom brisanju.  Primer:  [  {  " Apartman uspesno obrisan!"  }  ] |
| Format izlaznih parametara | application/json |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje musterije |
| HTTP metoda | POST |
| URL | http://MilanovicBNB.com/API/obrisi\_kupca?id=@id\_apartmana |
| URL parametri | INT id\_kupca npr 39 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Poruka o uspesnom/neuspesnom brisanju.  Primer:  [  {  Gost uspesno obrisan!"  }  ] |
| Format izlaznih parametara | application/json |
|  |  |
|  |  |

# Slučajevi korišćenja

### SK1 – Logovanje

**Naziv:** Logovanje

**Akteri:** Korisnik

**Preduslov:** Korisnik se nalazi u bazi

**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

**Alternativni scenario:**

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

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

### SK2 – Pregled apartmana u ponudi

**Naziv:** Pregled liste apartmana

**Akteri:**  Gost

**Osnovni scenario:**

1. Gost navigira do pocetne strane sajta
2. Gost pregleda listu apartmana

**Alternativni scenario:**

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

### SK3 – Brisanje apartmana

**Naziv:** Brisanje apartmana

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela
2. Administrator bira apartmana iz liste i pritiskom na dugme potvrđuje brisanje apartmana

**Alternativni scenario:**

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

### SK4 – Dodavanje apartmana

**Naziv:** Dodavanje apartmana

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela
2. Administrator upisuje podatke novog apartmana i pritiskom na dugme potvrđuje unos.

**Alternativni scenario:**

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

### SK5 – Pretraživanje apartmana

**Naziv:** Pretraživanje apartmana

**Akteri:** Admin

**Osnovni scenario:**

1. Admin navigira do strane za pretraživanje apartmana
2. Admin unosi podatke za pretragu i pritiskom na dugme vrši pretragu

**Alternativni scenario:**

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

### SK6 – Kupovina apartmana

**Naziv:** Kupovina apartman

**Akteri:** Gost

**Osnovni scenario:**

1. Gost lista artikle, bira zeljeni i klikom na KUPI pravi porudzbinu
2. Gost dobija obavestenje da ce apartman biti isporucen na njegovu adresu

**Alternativni scenario:**

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

### SK7 – Pretraživanje svojih Rezervacija

**Naziv:** Pretraživanje Rezervacija

**Akteri:** Gost

**Osnovni scenario:**

1. Gost navigira do strane Moje porudzbine
2. Gost unosi podatke relevantne podatke 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.

### SK8 – Pretraživanje svih Rezervacija

**Naziv:** Pretraživanje Rezervacija

**Akteri:** Administrator

**Osnovni scenario:**

1. Administrator navigira do strane upravljanje Rezervacijama
2. Administrator unosi podatke relevantne podatke 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 – Brisanje porudzbine

**Naziv:** Brisanje porucenih apartmana

**Akteri:** Admin

**Osnovni scenario:**

1. Admin navigira do svih Rezervacija
2. Admin upisuje podatke vezane za porudzbinu i pronalazi zeljenu porudzbinu
3. Klikom na OBRISI brise porudzbinu

**Alternativni scenario:**

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

### SK10 – Otkazivanje porudzbine

**Naziv:** Otkazivanje porudzbine

**Akteri:** Gost

**Osnovni scenario:**

1. Gost navigira do stranice za preged svojih Rezervacija – Moje porudzbine
2. Gost pritiskom na link za brisanje izabrane porudzbine briše istu sa liste .

**Alternativni scenario:**

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

### SK11 – Izmena podataka apartman

**Naziv:** Izmena podataka apartman

**Akteri:** Admin

**Osnovni scenario:**

1. Admin vrši pretragu apartmana i bira apartman čije podatke želi izmeniti.
2. Unosi podatke i pritiskom na dugme potvrđuje nove podatke o apartmanu.

**Alternativni scenario:**

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

### SK13 – Pretraga Gosta

**Naziv:** Pretraga Gosta

**Akteri:** Admin

**Osnovni scenario:**

1. Admin navigira na stranu za upravljanje kupcima.
2. Admin vrši pretragu Gosta po zadatom kriterijumu.

**Alternativni scenario:**

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

### SK14 – Pregled detalja apartman

**Naziv:** Pregled detalja apartman

**Akteri:** Admin

**Osnovni scenario:**

1. Admin pretražuje artikle i bira apartman čije podatke želi pregledati.
2. Admin bira zeljenog apartman i prikazuju mu se detalji o njemu

**Alternativni scenario:**

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

### SK16 – Logout

**Naziv:** Logout

**Akteri:** Korisnik

**Osnovni scenario:**

1. Korisnik navigira do stranoce za logout.
2. Korisnik se pritiskom na link za logout odjavljuje iz aplikacije.

**Alternativni scenario:**

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

#### SK17: Slučaj korišćenja – Registracija korisnika

**Naziv:** Registracija korisnika

**Svrha:** Registrovanje novog korisnika na sistem

**Akter:** Korisnik

**Preduslov:** /

**Osnovni scenario:**

1.Korisnik na pocetnoj strani popunjava formu za registraciju

2.Korisnik potvrđuje unos klikom na dugme registruj

3.Sistem unosi korisnika u bazu i prosleđuje korisnika na njegovu profil stranu

**Alternativni scenario:**

3.1 Ukoliko je korisnik propustio da unese neko polje sistem ga obavestava o tome

3.2 Ukoliko polje za e-mail adresu ne sadrži simbol ’@’, sistem obaveštava korisnika da e-mail adresa nije u odgovarajućem formatu

3.3 Ukoliko sistem ne može da se poveže na bazu pokazuje poruku o grešci

## 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



### DS2 – Pregled liste apartmana

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela
2. Administrator pregleda listu apartmana



### DS3 – Brisanje apartmana

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela
2. Administrator bira apartmana iz liste i pritiskom na dugme potvrđuje brisanje apartmana



### DS4 – Dodavanje apartmana

**Osnovni scenario:**

1. Administrator navigira do administratorskog panela
2. Administrator upisuje podatke novog apartmana i pritiskom na dugme potvrđuje unos.



### DS5 – Pretraživanje apartmana

**Osnovni scenario:**

1. Admin navigira do strane za pretraživanje Admina
2. Admin unosi podatke za pretragu i pritiskom na dugme vrši pretragu



### DS6 – Pretraživanje Gosta

**Osnovni scenario:**

1. Admin navigira do strane za pretragu Gosta
2. Admin unosi podatke relevantne podatke u obliku teksta i pritiskom na dugme izvršava pretragu.



### DS8 – Pretraživanje Rezervacija

**Osnovni scenario:**

1. Admin navigira do strane za upravljanje Rezervacijama
2. Admin unosi podatke relevantne podatke u obliku teksta i pritiskom na dugme izvršava pretragu.



### DS9 – Brisanje Gosta

**Osnovni scenario:**

1. Admin navigira do stranice za upravljanje kupcima
2. Admin pritiskom na link za brisanje izabranog Gosta briše isti sa liste .

****

### DS10 – Izmena podataka apartman

**Osnovni scenario:**

1. Admin vrši pretragu apartmana i bira apartman čije podatke želi izmeniti.
2. Unosi podatke i pritiskom na dugme potvrđuje nove podatke.

.



### DS11 – Pregled liste Rezervacija

**Osnovni scenario:**

1. Admin navigira do strane Upravljanje Rezervacijama
2. Admin koristi listu i aktivne linkove kako bi pregledao porudzbine



### SK12 – Logout

**Osnovni scenario:**

1. Korisnik navigira do stranoce za logout.
2. Korisnik se pritiskom na link za logout odjavljuje iz aplikacije.

