Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



Seminarski rad – Primena Chart.js-a na aplikaciji za zalihe garderobe

Profesor: Dušan Barać Studenti: Jovana Drobnjakovicć 22/16

Beograd, 2020.

Sadržaj

[1. Korisnički zahtev 1](#_Toc29754871)

[2. Opis sistema 2](#_Toc29754872)

[2.1. Opis slučajeva korišćenja 2](#_Toc29754873)

[2.1.1 Kačenje dokumenta 4](#_Toc29754874)

[2.1.2 Brisanje proizvoda sa zaliha(stanja) 6](#_Toc29754875)

[2.1.3 Izmena stanja 8](#_Toc29754876)

[3.1. Opis arhitekture aplikacije 11](#_Toc29754877)

[3.1.1 Dijagram klasa 11](#_Toc29754878)

[3.1.2 Struktura baze 12](#_Toc29754879)

[Baza se sastoji iz sest tabela: brend, prodavnica, proizvod, stanje, user,images 12](#_Toc29754880)

[4.1. Specifikacija REST API-ja 13](#_Toc29754881)

[3. Opis tehnologija korišćenih u aplikaciji 19](#_Toc29754882)

[3.1.HTML 19](#_Toc29754883)

[3.2.CSS 20](#_Toc29754884)

[3.3.PHP 21](#_Toc29754885)

[3.4. AJAX(Asynchronous JavaScript And XML) 22](#_Toc29754886)

[3.5.JSON 23](#_Toc29754887)

[3.6.JavaScript 24](#_Toc29754888)

[3.6.1. Chart.js 25](#_Toc29754889)

[3.7.Jquery 25](#_Toc29754890)

[4. Korisničko uputstvo - Opis karakterističnih slučajeva korišćenja preko screenshot-ova aplikacije 26](#_Toc29754891)

[4.2.Administrator 31](#_Toc29754892)

[5.Prikaz reprezentativnih delova koda 34](#_Toc29754893)

# Korisnički zahtev

Potrebno je projektovati i implementirati veb aplikaciju za zalihe garderobe-AJI zalihe. Sistem treba da prepoznaje dve vrste korisnika: administratora sistema i korisnika sistema sa običnim privilegijama. Za korisnika sistema je potrebno omogućiti registraciju na sajt i logovanje. Ako korisnik nije registrovan, nije moguće da se prijavi. Administrator sistema ima pristup administratorskom panelu. Administratorski panel se sastoji iz dela koji se odnosi na brisanje ili izmenu zaliha, dodavanje zaliha na stanje kao i dodavanje proizvoda. Zatim je omogućena pretraga proizvoda prema kategoriji, vizuelizacija sa prikazom grafikona gde nam je predstavljena kolicina po svakom proizvodu, kao i kačenje pdf fajlova na sajt.

# Opis sistema

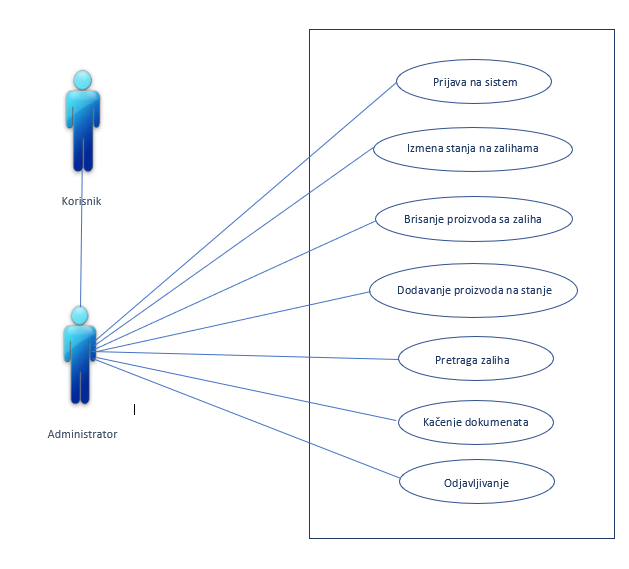
## 2.1. Opis slučajeva korišćenja

Karakteristični slučaji koriščenja:

1. Registracija korisnika
2. Prijava (login) korisnika/admina
3. Pretraga radnji po lokaciji
4. Pregled stanja
5. Dodaj na stanje
6. Dodaj proizvod
7. Izmena zaliha
8. Brisanje zaliha
9. Dodavanje slika
10. Dodavanje brenda
11. Kačenje dokumenata
12. Odjavljivanje korisnika/administratora

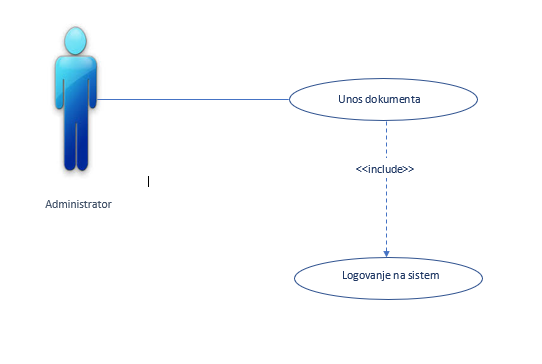


Dijagram slučajeva korišćenja 1 – Korisnik



Dijagram slučajeva korišćenja 2 – Administrator

## 2.1.1 Kačenje dokumenta



Dijagram slučajeva korišćenja 3 – Kačenje dokumenata

**Slučaj korišćenja – Kačenje dokumenata**

***Naziv SK:* Kačenje dokumenta**

***Aktori SK:*** Administrator

***Učesnici SK:*** Administrator, sistem

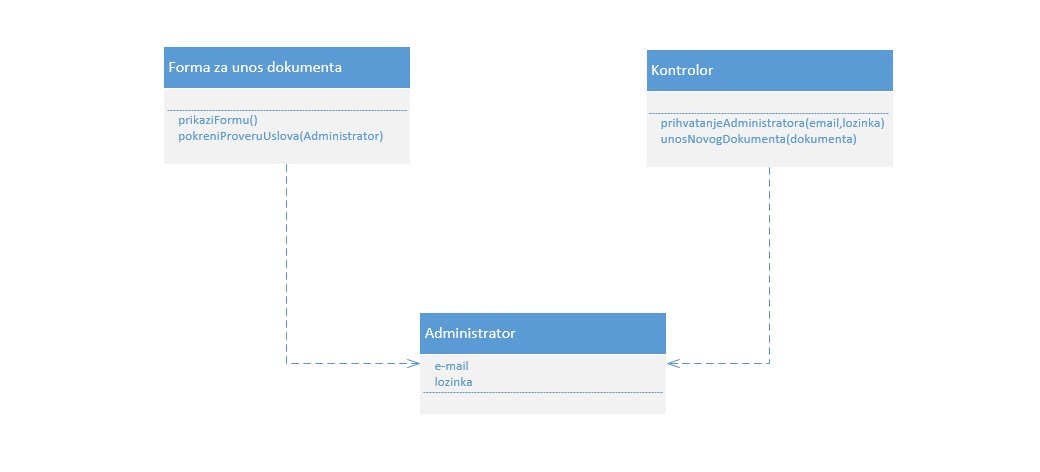
***Preduslov:*** Sistem je pokrenut.

***Osnovni scenario:***

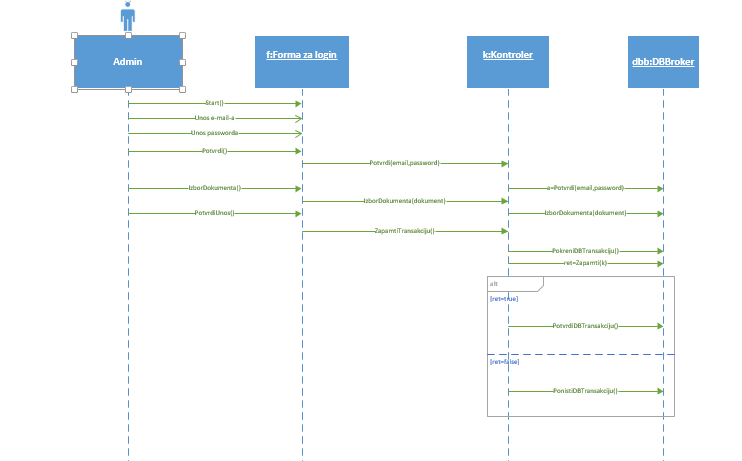
1. Administrator se loguje na sistem.
2. Korisnik šalje zahtev sistemu za logovanje.
3. Sistem prihvata korisnika.
4. Administrator unosi dokument
5. Sistem čuva dokument.

***Alternativni scenario:***

* 1. Ukoliko sistem ne moze da prihvati logovanje, šalje poruku korisniku: “Neuspešno logovanje.“



UML dijagram 1 Kačenje dokumenata



Dijagram sekvenci 1 Kačenje dokumenta

## 2.1.2 Brisanje proizvoda sa zaliha(stanja)

**Slučaj korišćenja – Brisanje proizvoda sa zaliha**

***Naziv SK:*** Brisanje proizvoda sa zaliha

***Aktori SK:*** Administrator

***Učesnici SK:*** Administrator i sistem

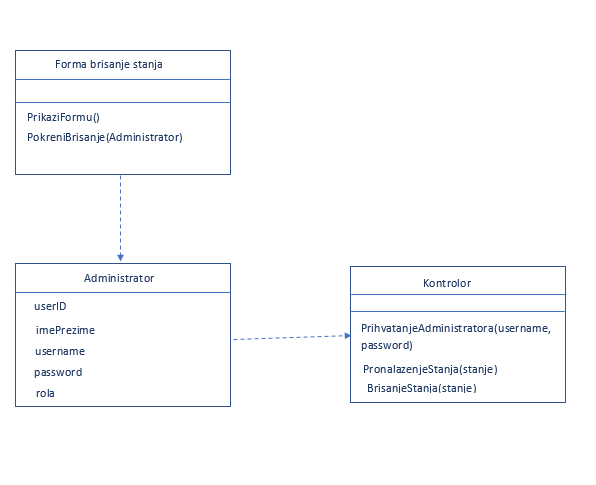
***Preduslov:*** Sistem je pokrenut

**Osnovni scenario:**

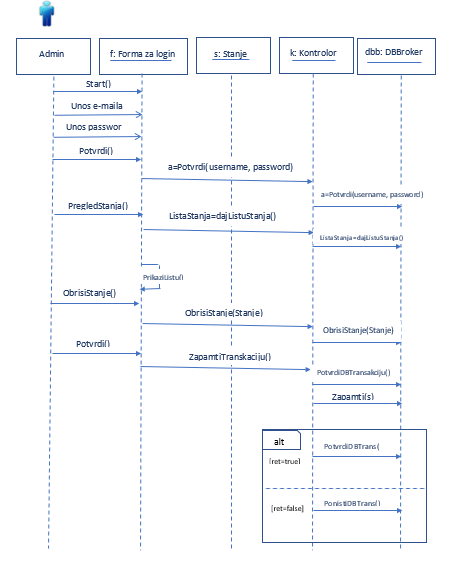
1. Administrator unosi podatke za logovanje.
2. Administrator poziva sistem da izvrsi logovanje.
3. Sistem izvrsava logovanje.
4. Nakon uspešnog logovanja administrator se nalazi na početnoj strani.
5. Administrator pronalazi proizvod koji želi da ukloni.
6. Administrator pritiskom na x poziva sistem da obriše stanje.

**Alternativni scenario**

3.1Ukoliko sistem ne moze da uloguje administratora, salje poruku: “Neuspesno logovanje.”



UML dijagram 2 Brisanje stanja



Dijagram sekvenci 2 Brisanje stanja

## 2.1.3 Izmena stanja

**Slučaj korišćenja – Izmena stanja**

***Naziv SK:*** Izmena stanja

***Aktori SK:*** Administrator

***Učesnici SK:*** Administrator i sistem

***Preduslov:*** Sistem je pokrenut.

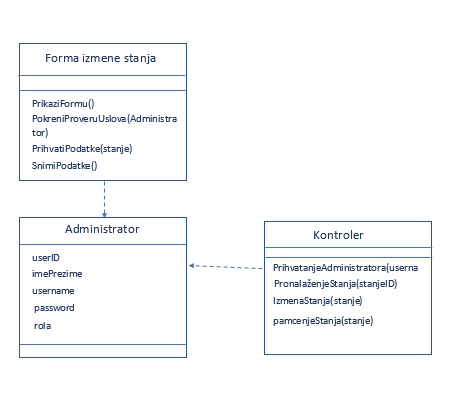
**Osnovni scenario:**

**Osnovni scenario:**

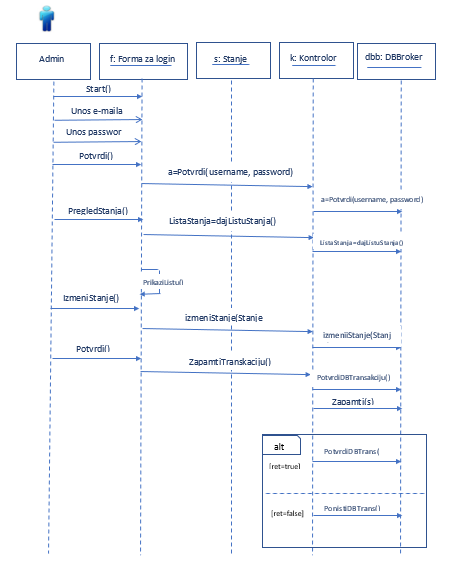
1. Administrator unosi podatke za logovanje.
2. Administrator poziva sistem da izvrsi logovanje.
3. Sistem izvrsava logovanje.
4. Nakon uspešnog logovanja administrator se nalazi na početnoj stranici.
5. Administrator pronalazi stanje koji želi da izmeni.
6. Administrator pritiskom na oznaku za izmenu poziva sistem da izmeni stanje.

**Alternativni scenario**

3.1 Ukoliko sistem ne može da uloguje administratora, šalje poruku korisniku: “Neuspešno logovanje.”

****

UML dijagram klasa 2 Izmena stanja



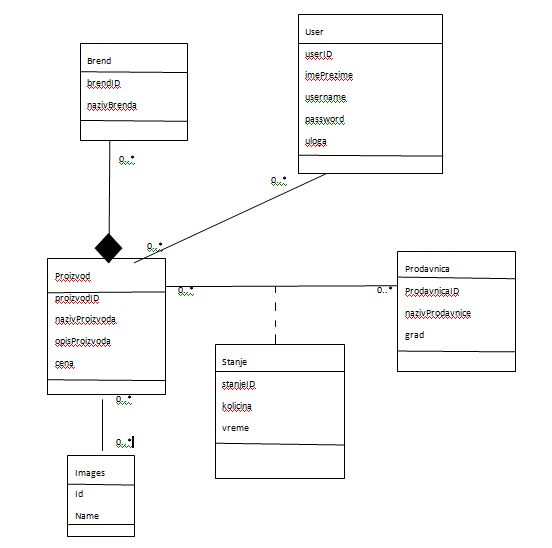
Dijagram sekvenci 3 Izmena stanja

## 

## 3.1. Opis arhitekture aplikacije

## 3.1.1 Dijagram klasa

UML dijagram Arhitektura sistema

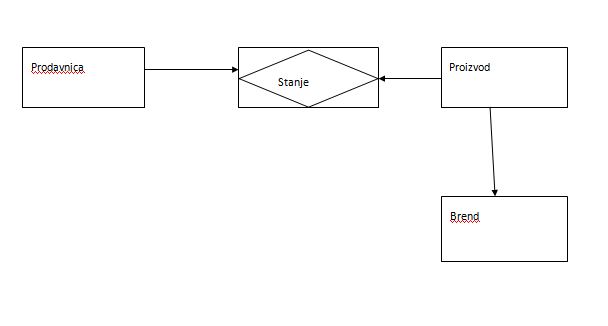


## 

## 3.1.2 Struktura baze

U radu je koriščena baza podataka ajizalihe.

## Baza se sastoji iz sest tabela: brend, prodavnica, proizvod, stanje, user,images



PMOV za bazu

Tabela Prodavnica ima sledeće kolone: ProdavnicaID, nazivProdavnice, grad. Atribut ProdavnicaID je primarni ključ tabele. Tabela Proizvod sadrži: ProizvodID, naziv, cena, opis, kao i ProdavnicaID koji predstavlja spoljni ključ ove tabele. ProizvodID je primarni ključ. Brend čine kolone: brendID (primarni ključ), nazivBrenda. Tabela stanjezaliha koja je prestavlja agregaciju između Prodavnice i Proizvoda sadrži: StanjeID koji je primarni ključ, spoljne ključeve ProdavnicaID i ProizvodID kao i količinu I vreme. Tabela User ima sledeće kolone: UserID koja predstavlja primarni ključ, imePrezime, username, password I kolonu uloga. . Uloga može uzimati vrednosti 0 ili 1, pri čemu na taj način definiše da li je pristup omogućen korisniku ili administrator.

## 4.1. Specifikacija REST API-ja

|  |  |
| --- | --- |
| Opis funkcije | Pregled svih korisnika |
| HTTP metoda | GET |
| URL | api/korisnici |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribute userID[int], imePrezime[string], password[string], uloga[string], usernamestring]  Primer:  [  {  "userID":"1",  "imePrezime":"Maja Radosavljevic",  "username":"maki995",  "password":"makica",  "uloga":"1"  },  {  "userID":"2",  "imePrezime":"Danijela Cukavac",  "username":"Daca995",  "password":"0",  "uloga":"0"  },  {  "userID":"3",  "imePrezime":"Ana Markovic",  "username":"ancianci",  "password":"ancika",  "uloga":"0"  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Pregled stanja |
| HTTP metoda | GET |
| URL | api/stanja |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima stanjeID[int], proizvodID[int],prodavnicaID[int], kolicina[int],vrene[datetime],nazivProdavnice[string], grad[string],nazivproizvoda[string],opisProizvoda[string[, cena[double],brendID[int], nazivBrenda[string]  Primer:  [  {  "stanjeID":"1",  "proizvodID":"2",  "prodavnicaID":"1",  "kolicina":"20",  "vreme":"2018-02-14 18:34:30",  "nazivProdavnice":"Olja Butik",  "grad":"Arilje",  "nazivProizvoda":"Karirana suknja",  "opisProizvoda":"Najnoviji model u kolekciji prolece 2018.",  "cena":"3200",  "brendID":"3",  "nazivBrenda":"H&M"  },  {  "stanjeID":"2",  "proizvodID":"1",  "prodavnicaID":"1",  "kolicina":"10",  "vreme":"2018-02-14 18:35:05",  "nazivProdavnice":"Olja Butik",  "grad":"Arilje",  "nazivProizvoda":"Haljina na pruge",  "opisProizvoda":"Kupite po najboljoj ceni jedinstvenu zarinu haljinu!",  "cena":"3000",  "brendID":"1",  "nazivBrenda":"Zara"  },  {  "stanjeID":"3",  "proizvodID":"3",  "prodavnicaID":"3",  "kolicina":"5",  "vreme":"2018-02-15 01:17:04",  "nazivProdavnice":"Fashion Butik",  "grad":"Novi Sad",  "nazivProizvoda":"Bela kosulja",  "opisProizvoda":"Klasicni model za sve prilike",  "cena":"1200",  "brendID":"2",  "nazivBrenda":"Bershka"  },  {  "stanjeID":"4",  "proizvodID":"3",  "prodavnicaID":"2",  "kolicina":"7",  "vreme":"2018-02-15 01:22:06",  "nazivProdavnice":"Extreme Intimo",  "grad":"Beograd",  "nazivProizvoda":"Bela kosulja",  "opisProizvoda":"Klasicni model za sve prilike",  "cena":"1200",  "brendID":"2",  "nazivBrenda":"Bershka"  },  {  "stanjeID":"5",  "proizvodID":"2",  "prodavnicaID":"3",  "kolicina":"22",  "vreme":"2018-02-15 01:22:35",  "nazivProdavnice":"Fashion Butik",  "grad":"Novi Sad",  "nazivProizvoda":"Karirana suknja",  "opisProizvoda":"Najnoviji model u kolekciji prolece 2018.",  "cena":"3200",  "brendID":"3",  "nazivBrenda":"H&M"  },  {  "stanjeID":"7",  "proizvodID":"1",  "prodavnicaID":"4",  "kolicina":"5",  "vreme":"2018-02-15 14:58:58",  "nazivProdavnice":"Selection",  "grad":"Zrenjanin",  "nazivProizvoda":"Haljina na pruge",  "opisProizvoda":"Kupite po najboljoj ceni jedinstvenu zarinu haljinu!",  "cena":"3000",  "brendID":"1",  "nazivBrenda":"Zara"  },  {  "stanjeID":"8",  "proizvodID":"5",  "prodavnicaID":"2",  "kolicina":"60",  "vreme":"2018-02-15 19:35:07",  "nazivProdavnice":"Extreme Intimo",  "grad":"Beograd",  "nazivProizvoda":"Sivi kaput",  "opisProizvoda":"Jesenja kolekcija",  "cena":"5000",  "brendID":"5",  "nazivBrenda":"Stradivarius"  },  {  "stanjeID":"9",  "proizvodID":"4",  "prodavnicaID":"2",  "kolicina":"20",  "vreme":"2018-02-17 19:19:36",  "nazivProdavnice":"Extreme Intimo",  "grad":"Beograd",  "nazivProizvoda":"Tunika na pruge",  "opisProizvoda":"Najnoviji model kolekcije prolece\/leto.",  "cena":"1500",  "brendID":"9",  "nazivBrenda":"Guess"  }  ] |
| Format HTTP body parametara | nema |
| Izlazni parametri | nema |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Kategorije |
| HTTP metoda | GET |
| URL | api/brend |
| URL parametri | (nema) |
| HTTP body parametri | [  {  "brendID":"1",  "nazivBrenda":"Zara"  },  {  "brendID":"2",  "nazivBrenda":"Bershka"  },  {  "brendID":"3",  "nazivBrenda":"H&M"  },  {  "brendID":"4",  "nazivBrenda":"ExtremeIntimo"  },  {  "brendID":"5",  "nazivBrenda":"Stradivarius"  },  {  "brendID":"7",  "nazivBrenda":"Beba Kids"  },  {  "brendID":"8",  "nazivBrenda":"Sport Vision"  },  {  "brendID":"9",  "nazivBrenda":"Guess"  },  {  "brendID":"10",  "nazivBrenda":"Replay"  },  {  "brendID":"11",  "nazivBrenda":"Disquared"  },  {  "brendID":"12",  "nazivBrenda":"Nike"  },  {  "brendID":"13",  "nazivBrenda":"Puma"  },  {  "brendID":"14",  "nazivBrenda":"Adidas"  },  {  "brendID":"15",  "nazivBrenda":"Champion"  },  {  "brendID":"16",  "nazivBrenda":"Champion"  }  ] |
| Format HTTP body parametara | nema |
| Izlazni parametri | nema |
| Format izlaznih parametara | application/json |

# 3. Opis tehnologija korišćenih u aplikaciji

U našem radu koristili smo sledeće tehnologije:

1. HTML
2. CSS
3. PHP
4. AJAX
5. JSON
6. JavaScript
7. XML
8. Jquery

## 3.1.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 računarski 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 koriste se sledeće tehnologije:

* 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.

Svaki HTML dokument mora početi tagom <html> i završiti se tagom </html>. HTML dokument sadrži uobičajeno dva različita dela, head i body. Head sadrži informacije o dokumentu koje se ne prikazuju na ekranu. Body sadrži sve ostalo što se prikazuje kao deo web stranice.

HTML5 je jezik koji se koristi za strukturiranje i prezentovanje sadržaja za World Wide Web i predstavlja osnovnu tehnologiju Interneta. HTML5 je peta revizija HTML standarda koja se i dalje dopunjuje i razvija. Osnovni ciljevi razvijanja ove poslednje verzije HTML-a su poboljšanje jezika sa podrškom za multimedije koji će biti razumljiv za ljude, ali isto tako i za kompjutere i uređaje (web pretraživače, parsere...). HTML5 je potencijalni kandidat za „cross-platform“ mobilne aplikacije pošto su mnoge osobine HTML5 pravljene tako da budu u mogućnosti da se pokrenu na uređajima male snage kao što su smart telefoni i tableti.

HTML5 je namenjen da prevaziđe ne samo HTML4, nego XHTML1 i HTML DOM. WHATWG je radio na web formama i aplikacijama, dok je W3C radio na XHTML 2.0 . 2006. godine su počeli zajedničku saradnju. Tako da se HTML5 može smatrati mešavinom karakteristika i specifikacija HTML-a i XHTML-a. Ovome je doprinela zajednička praksa, kao i mnoge greške u postojećim web dokumentima, ovo je takođe i pokušaj da se definiše jedinstveni markup jezik, koji se moţe pisati i u HTML-u i u XHTML-u. Ovo uključuje detaljne procesne modele da bi ohrabrilo još interoperabilnih implementacija. To proširuje, unapređuje i racionalizuje označavanje dokumenata, kao što je omogućen i API (application programming interfaces), za kompleksne web aplikacije. Trenutno postoje dve paralelne verzije HTML 5, jedna sluţbena na kojoj radi W3C i nesluţbenu na kojoj rade stručnjaci iz Applea, Mozille, Opere i Googlea, koji imaju velik uticaj na razvoj Interneta, iako iza sebe nemaju formalnu organizaciju kao što je W3C. Ipak, urednici HTML 5 specifikacije za oba tima su isti ljudi, što znači da grupe međusobno saraĎuju i da ćemo na kraju ipak imati jedan standard. Urednici HTML 5 specifikacije su Ian Hickson iz Googlea i David Hyatt koji radi za Apple.

HTML5 uvodi mnoge nove sintaksne promene. One uključuju nove <video>, <audio> i <canvas> elemente, kao i integraciju SVG (Scalable Vector Graphics) sadržaja (koji zamenjuju korišćenje <object> tagova). Ove karakteristike su dizajnirane tako da lako uključuju i rukuju multimedijalnim i grafičkim sadržajem na web-u bez potrebe uključivanja plugin-ova i API-ja. Drugi novi elementi, kao što su na primer <section>, <article>, <header> i <nav> su kreirani da obogate semantički sadržaj dokumenta. Sa druge strane, neki elementi i atributi su uklonjeni, kao na primer <font> i <center> koji su prevaziđeni korišćenjem mnogo moćnijeg css-a.

## 3.2.CSS

Iako koristi da se njime opisuju web stranice i korisnički interfejsi napisani u HTML-u i XHTML-u, CSS-om se takođe mogu opisati bilo koje vrste XML dokumenata uključujući i sam XML, SVG ili XUL. Uz korišćenje HTML i Javascript jezika, CSS daje vizuelnu strukturu i opis web sajtova, web aplikacija kao i aplikacija za mobilne telefone.

Tri osnovne karakteristike CSS jezika su mogućnost za definisanje klasa za izgled, boje i fontova. Ovi elementi omogućavaju pristupačniji i fleksibilniji sadržaj kao i kontrolu web dizajnera nad određenom grupom HTML elemenata u sadržaju. Na primer, znamo da HTML ima pojedine tagove kojima mogu da se definišu pojedini elementi, kao što je slučaj sa tagom <bold>. On omogućava podebljanost teksta na stranici. Ukoliko želimo da nam svaki naslov bude podebljan, korišćenjem CSS-a ćemo izbeći konstantno ponavljanje <bold> tagova na svakom mestu gde se nalazi naslov tako što ćemo definisati klasu za izgled <h1> tagova unutar CSS-a i samo pozvati tu klasu prilikom ispisivanja naslova.

CSS3 je najnoviji standard CSS-a. CSS3 specifikacija je i dalje u razvoju od strane W3C organizacije. Većina css3 svojstava je implementirana u modernim web pretraživačima.

CSS3 je podeljen na module. Stara specifikacija je podeljena na manje delove i dodati su novi delovi. Neki od najvažnijih modula su:

* Box model
* Backgrounds i borders
* Tekstualni efekti i web fontovi
* 2D/3D transformacije
* Animacije
* Multiple column layout
* User interface
* Selektori

## 3.3.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.

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 smislu da dozvoljava i čisto-proceduralno programiranje ali omogućava i korišćenje klasa i drugih koncepata objektno orijentisanog programiranja (nasleđivanje, apstraktne metode, interfejsi itd.).

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*”
* Program koji se napiše u PHP-u ne zahteva prevođenje (kompajliranje), nego se interpretira pri svakom izvršavanju. PHP interpreter može raditi po CGI principu, odnosno tako što će interpreter 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 veb-servisa. Druga varijanta je danas u najvećoj upotrebi jer pruža znatno veću brzinu izvršavanja - interpreter je na taj način uvek učitan u memoriju te se ne mora pozivati spoljašnji program.

Uobičajen scenario po kom se izvršavaju PHP skripte na web serveru je sledeći:

* klijent (korisnik Interneta koji koristi neki brauzer) zahteva PHP stranicu sa servera
* server prosleđuje zahtev servisu za veb (program veb-server na serveru)
* veb-server prepoznaje da se traži PHP datoteka
* ne šalje njegov sadržaj klijentu, nego ga izvršava kao program pomoću PHP modula
* izlazni tekst programa (standardni izlaz) se šalje klijentu kao rezultat zahteva
* klijent prepoznaje vrstu rezultata (HTML kod, slika, PDF sadržaj, arhiva itd.)

rezultat se prikazuju klijentu na odgovarajući način

## 3.4. AJAX(Asynchronous JavaScript And XML)

AJAX je skraćenica od “Asynchronous JavaScript and XML”. Termin AJAX prvi put je upotrebljen u februaru 2005. godine, kada je Džesi Džejms Garet (Jesse James Garret), dizajner informacionih sistema i direktor kompanije Adaptive Path, pokušao da nađe odgovarajuću skraćenicu za grupu tehnologija koju je predlagao svom klijentu. Tehnologije potrebne za postojanje AJAX -a su postojale i ranijih godina ali je glavni razlog za slabiju upotrebu svih tehnika korišćenih u AJAX-u, nedostatak podrške od strane web čitača. Mnogi su godinama pre nje koristile slične principe, poput Microsoftovog Remote Scriptinga ili veoma raširenog DHTML-a. Najbitnija stvar koja je uticala na brz razvoj AJAX-a jeste trenutak na tržištu u kojem su veliki igrači Internet industrije želeli da težište korišćenja računara prenesu sa desktopa na web stranice, za šta im je bila potrebna upravo tehnologija.

Web aplikacije imaju brojne prednosti u odnosu na desktop aplikacije. Web aplikacije dosežu do većeg broja ljudi, lakše se razvijaju, implementiraju i održavaju.

Najveći nedostatak Web aplikacija u odnosu na desktop aplikacije je taj što su desktop aplikacije „bogatije” u smislu funkcija i informacija koje nude. Međutim, taj nedostatak je u velikoj meri otklonjen uz pomoć AJAX tehnologije.

Web aplikacije su one aplikacije čijom funkcionalnošću se upravlja preko web servera i dostavljaju se korisnicima preko mreže kao što je internet ili intranet. I pored mnogostrukih prednosti, klasične web aplikacije imaju određene neostatke, kada je u pitanju interakcija sa korisnicima:

* Spor odgovor na zahteve
* Gubitak sadržaja prilikom ponovnog učitavanja stranice
* Gubitak informacija na ekranu
* Gubitak pozicije skrolovanja
* Bez trenutnog odgovora na korisničke akcije
* Korisnik mora da čeka na učitavanje sledeće strane
* Klikni, čekaj, refresh princip rada na stranici
* Stranica se ponovno učitaca sa servera za sve akcije korisnika
* Sinhroni način rada.

Pre svega, svaki put kada se učitava nova stranica, postoji određeno vreme kada aplikacija «stoji». Veoma često se prilikom malih izmena ili zahteva za delovima stranice, ponovo učitava cela stranica, iako je najveći deo nove stranice u potpunosti identičan sa prethodnom.

Da bi se shvatio koncept AJAX tehnologija, neophodno je objasniti razliku izmenu sinhronog i asinhronog prenosa podataka između klijenta i servera.

U sinhronom načinu rada procesi se izvode sekvencijalno. Između dva procesa postoji jaz koji je jednak vremenu izvođenja drugog procesa. Dakle, osnovni proces komunikacije između klijenata i servera može se opisati na sledeći način: klijent šalje zahtev ka serveru, podaci se prenose ka serveru, server obrađuje podatke I zatim ih vraća klijentu. Ono što je primetno sa slike dole je da postoji vremenski interval između pravljenja zahteva i odgovora na isti. Za vreme tog intervala klijent čeka, bez mogućnosti da pravi nove zateve ili nastavi svoje korišćenje web aplikacije. Može se zaključiti da kod sinhronog prenosa podataka ne postoji mogućnost za eliminisanje čekanja zato što se procesi obavljaju jedan za drugim, čak i ukoliko za to nema potrebe.

## 3.5.JSON

JSON (JavaScript Object Notation) predstavlja tekstualni format za serijalizaciju podataka. Izveden je od formata za zapis objekta skriptnog jezika JavaScript, međutim danas ovaj format koristi većina savremenih programskih jezika. Osnovna svrha JSON-a je prenos podataka i može se reći da predstavlja svojevrsnu alternativu XML-u. JSON je u memorijskom smislu manje zahtevan od XML-a, što znači da JSON fajl sa određenim podacima zauzima manje memorijskog prostora od XML fajla sa istim podacima.

JSON je zasnovan na JavaScript standardu Standard ECMA-262 3rd Edition - December 1999. Iako je JSON usko povezan sa JavaScript programskim jezikom, danas praktično svi programski jezici podržavaju JSON zapis podataka. Među njima su JavaScript, PHP, Java, C, C++, Pyhon, Ruby i drugi.

JSON se zasniva na dve strukture:

* Zbirka parova ime / vrednost. Na raznim jezicima, to je realizovano kao objekat, zapis, struktura, rečnik, heš tabela, lista sa ključevima ili asocijativni niz.
* Uređena lista vrednosti. U većini jezika, ovo je realizovana kao niz, vektor, lista ili sekvenca.

JSON format razlikuje više tipova podataka. Promenljive se u JSON-u nazivaju vrednosti. Od prostih tipova podataka, to su string, number, boolean i null, a od složenih tipova u pitanju su object i array.

## 3.6.JavaScript

JavaScript je objektno zasnovan skriptni jezik. Uključujemo ga u web stranicu da bi je učinili dinamičnijom. HTML (osnovni kod web stranica) se koristi samo za oblikovanje i uređivanje elemenata stranice (tekst, forme, linkove i tabele), ali nema šanse da diktiramo kako će se ti elementi ponašati. Mogućnost uključenja JavaScript skripte daje nam mnogo veću kontrolu kako se web stranica ponaša. Kombinovan sa HTML-om i CSS-om JavaScript čini DHTML (Dinamic HTML).

Objektno je zasnovan jer programer ne definiše samo tip podataka, nego i vrstu operacija (funkcija) koje se mogu primjeniti na strukture podataka. Na ovaj način, struktura podataka postaje objekat koji uključuje i podatke i funkcije. Pored toga, programeri mogu da kreiraju odnose između jednog i drugog objekta. Na primer, objekti mogu da pridobiju karakteristike od drugih objekata.

Skriptni je jezik jer se sastoji od serije komandi koje se očitavaju u interpreteru (program prevodioc), a da se predhodno ne kompajlira sadržaj (compiler- program prevodioc). Odnosno ne prevodi se u mašinski jezik (binarni kod- 1 i 0) iz koga nikada nećemo saznati originalni jezik, nego se komande direktno "čitaju" iz koda (source code ili bytecode). Zbog ove karakteristike JavaScript se izvršava na strani korisnika (client side), tj. na računaru na kojem je pokrenut sadržaj sa JavaScript-om. Sam po sebi, HTML dozvoljava posjetocu da pošalje podatke ka serveru na obradu. Nažalost ako ti podaci nisu validni cijeli proces se mora ponoviti sve dok se ne unesu validni podaci. Ovo je jedan od osnovnih razloga nastanka JavaScript-a koji provjerava vjerodostojnost podataka na klijentovom pregledniku (browser-u) i tako olakšava posao na web-u.

Rani skript jezici su se često nazivali batch jezici. Ovo su neki skrip jezici : ASP, JSP, PHP, Perl, Tcl, Python itd.

JavaScript je najpopularniji skriptni jezik na Internetu kojeg podržavaju svi poznatiji preglednici (Internet Explorer, Mozilla Firefox, Netscape, Opera). Evo nekoliko stvari koje je svaki prosječan surfer na internetu vidio: padajući meni, neobični pokazivač miša, iskakajući prozor, sat ... Međutim postoje ozbiljnije primjene JavaScript-a kao što su:

Detekcija preglednika -Određuje se tip preglednika koji se koristi pri pregledu vaše web stranice. Zavisno od preglednika, može se prikazati drugačija stranica posebno dizajnirana za taj preglednik.

Kontrola prozora -mogu se kontrolisati dimenzije, meniji, dugmadi, vrijeme prikaza itd.

Web kolačići (cookies) -pohranjeni podataci na korisnikovom računaru, koji se automatski preuzimaju sljedeci put kada korisnik posjeti vašu web stranicu

Provera sadržaja -dobar primjer za ovo je provjeravanje da li je korisnik upisao znak @ u polje za upis email adrese.

### 3.6.1. Chart.js

Biblioteka chart.js je jednostavna i prilagodljiva *JavaScript* biblioteka pogodna dizajnerima i programerima za kreiranje raznih vrsta grafikona, dijagrama i sličnih vizuelnih predstavljanja podataka . Ovo je jedna od najjednostavnijih *JavaScript* biblioteka, a ima i ugrađene mogućnosti animacije. Veoma se lako podešava i prilagođava korisiniku, a ima veoma dobru podršku od svih otvorenih projekata.

## 3.7.Jquery

jQuery je biblioteka čija je svrha da olakša upotrebu JavaScript-a na vašim web stranicama.

jQuery slogan „write less, do more“ dosta govori – sa jednom linijom jQuery koda možete pozvati funkcionalnosti za koje bi vam trebalo mnogo više rada i truda, ako bi koristili klasični JavaScript metod kodiranja.

jQuery obuhvata mnoge zadatke koji zahtevaju veliki broj linija JavaScript-a i omotava ih u metode koje možete pozvati sa samo jednom linijom.

jQuery takođe pojednostavljuje i druge komplikovane stvari u JavaScript-u kao što su AJAX i DOM manipulacija.

JQUERY BIBLIOTEKA SADRŽI SLEDEĆE FUNKCIONALNOSTI:

* HTML/DOM manipulacija
* CSS manipulacija
* HTML event metodi
* Efekti i animacije
* AJAX

**Dodatno:**[jQuery poseduje plugin-e](http://www.popwebdesign.net/popart_blog/2013/08/najbolji-jquery-dodaci/) sa kojima možete postići gotovo svaki zahtev koji zamislite.

Najvažniji razlog tolike poularnosti jQuery biblioteke je ekstezivnost. Veliki broj najvećih svetskih kompanija kao što su Google, Microsoft, IBM ga koriste.

Tri stvari su značajne i izdvojile su jQuery u odnosu na druge konkurente:

* korišćenje CSS selektora
* unobtrusive (nenametljiv) JavaScript
* jednostavnost upotrebe

jQuery se pojavio u vreme kada su se front-end developeri navikavali na kreiranje web stranica sa boljim semantičkim obeležavanjem, a učenje CSS-a je bilo zastupljeno širom web-a. jQuery koristi potpuno iste [CSS selektore](http://www.popwebdesign.net/popart_blog/2013/08/10-korisnih-css-selektora/) koje developeri koriste za stilizovanje web stranica da bi dodali ponašanje i nema potrebe za promenom konteksta kada se referencira na iste elemente. „Unobtrusive JavaScript“je metodologija koja definiše napredan način inplementiranja JavaScript koda na web stranicama. Suština prezentacije. Na ovaj način izbegavaju se standardni problemi JavaScript kodiranja kao što su nekonzistentnost na browser-ima i nedostatak skalabilnosti.

jQuery je napravljen tako da bude krajnje jednostavan za developere koji ga koriste.

# 4. Korisničko uputstvo - Opis karakterističnih slučajeva korišćenja preko screenshot-ova aplikacije

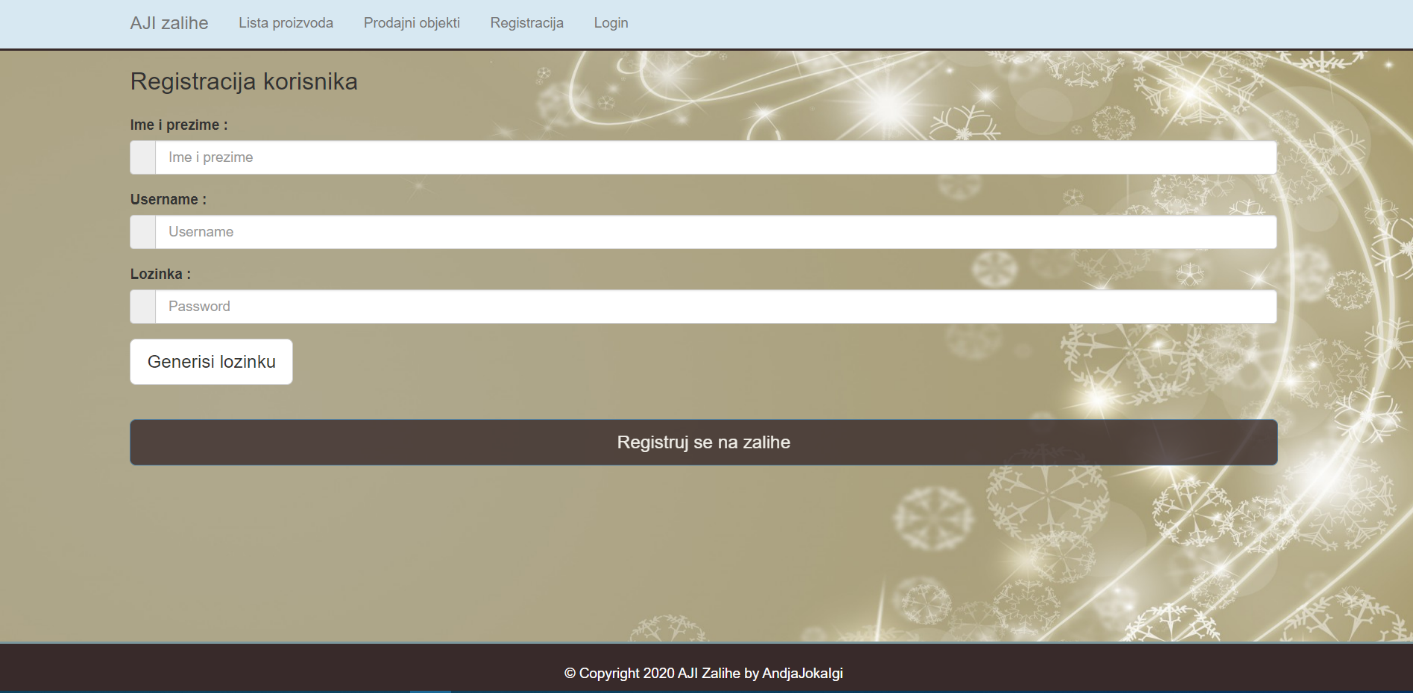


Početna

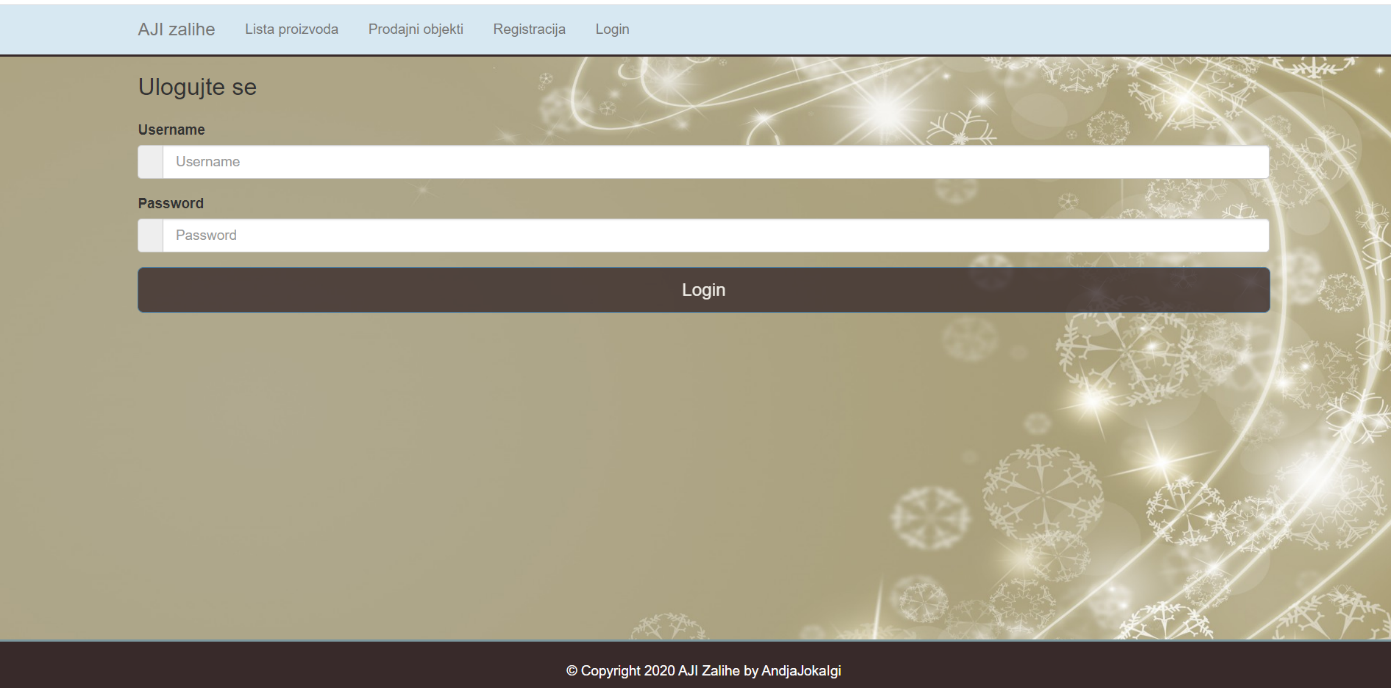


Meni

Klikom na karticu Registracija korisnik se registruje. Unosi imePrezime, username i password pomoću. Korisnik ima mogućnost da mu se lozinka sama izgeneriše. Generisana lozinka se dobija pritiskom na dugme “generiši lozinku” I pojavljuje se u vidu alert prozora.



Registracija

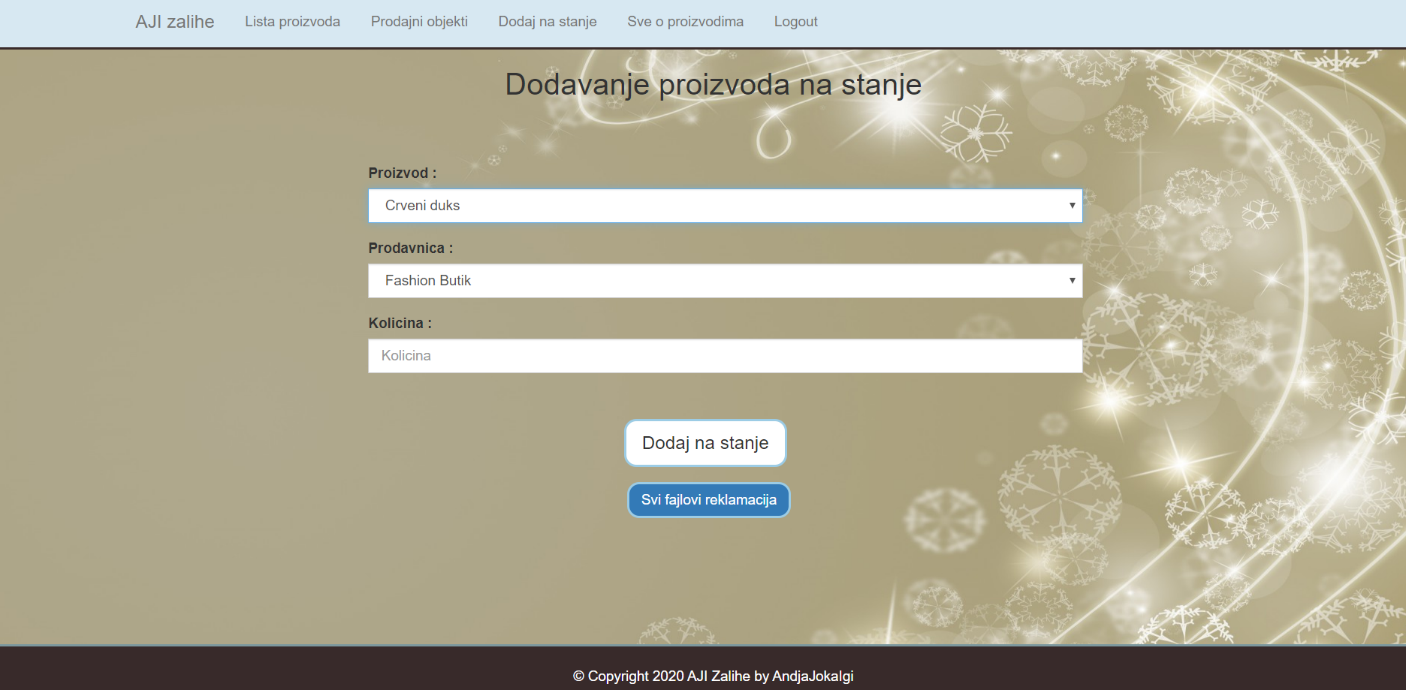


Login forma

Nakon uspešne prijave korisnike formira se novi menu.



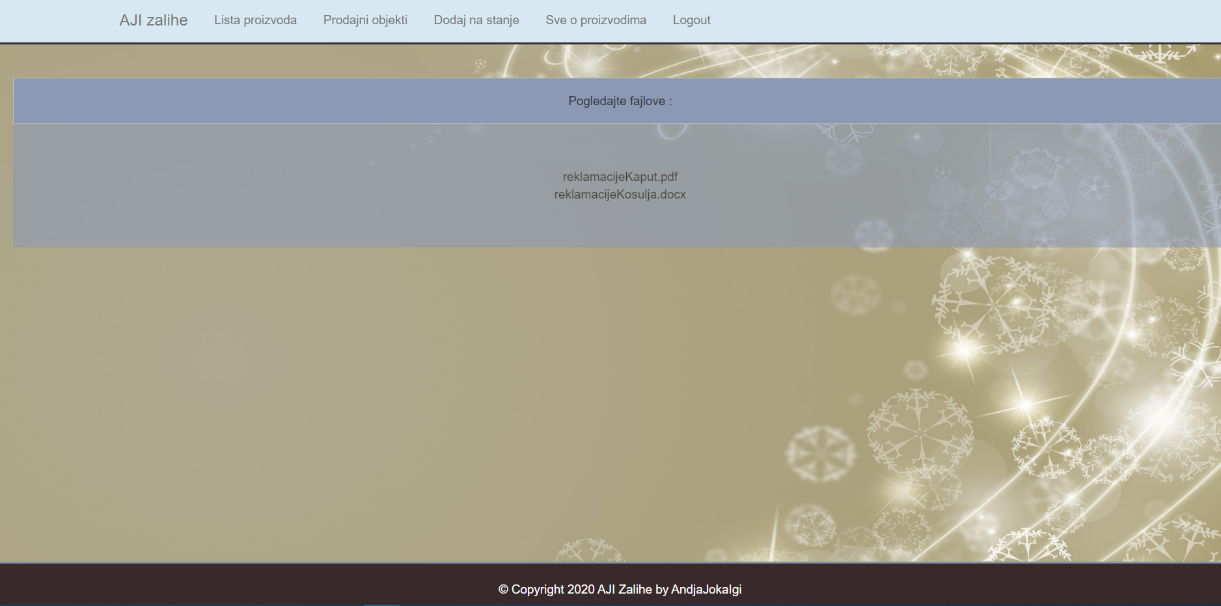
Menu obican korisnik



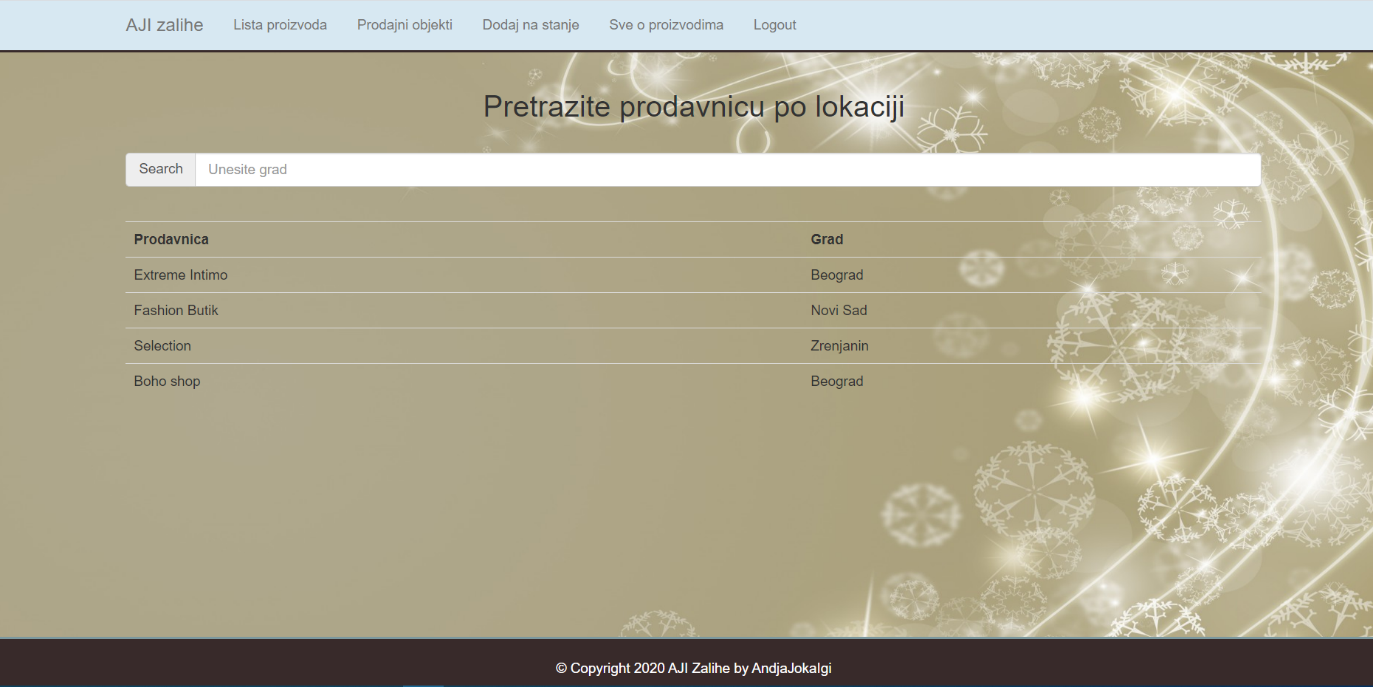
Dodavanje proizvoda na stanje

Korisnik može da doda proizvod na stanje. Prilikom dodavanja potrebno je da izabere prodavnicu, da odabere proizvod koji žele da unese kao i količinu.

Klikom na karticu „Svi fajlovi reklamacija“ korisnik može preuzeti neki od korisnih fajlova.



Fajlovi



Lokacije prodajnih objekata

Korisnik moze da pretrazi trenutna prodajna mesta Aji Zaliha.



Statistika proizvoda

Korisnik moze da vidi statistiku trenutnog stanja proizvoda na zalihama kroz grafik.

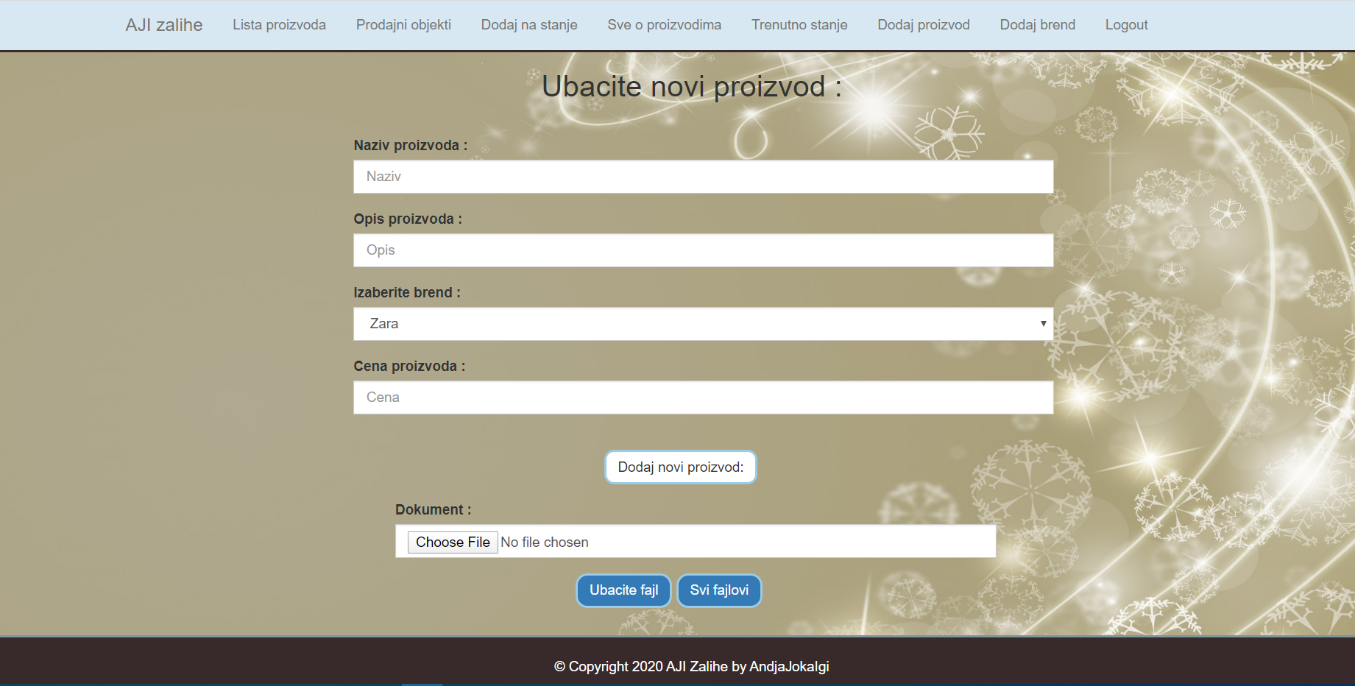
## 4.2.Administrator

Meni nakon što se korisnik uloguje kao admin. Omogućeno mu je da izmedna i brisanje proizvoda na stanju kao i dodavanje novog proizvoda.



Menu admin

Klikom na karticu „Dodaj proizvod“ prikazuje se sledeća forma.



Dodaj proizvod

Kartica „Izmena i brisanje “ je najznačajnija. Administrator vrši izmenu, brisanje, pretragu proizvoda na stanju kao i kačenje fajlova.



Izmena i brisanje



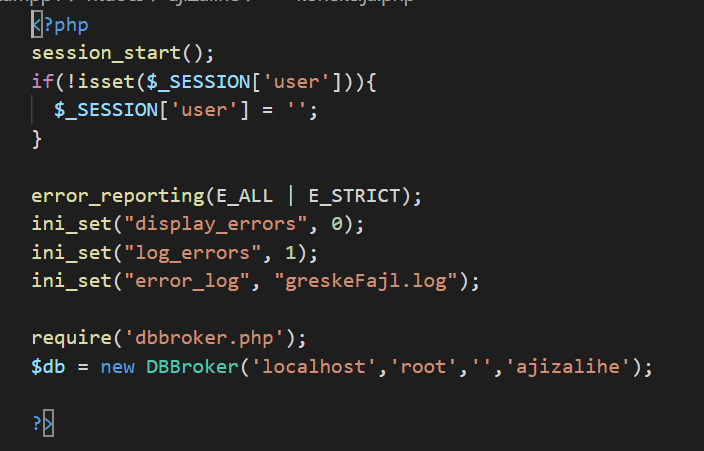
Izmena kolicine proizvoda na zalihama

# 

# 5.Prikaz reprezentativnih delova koda

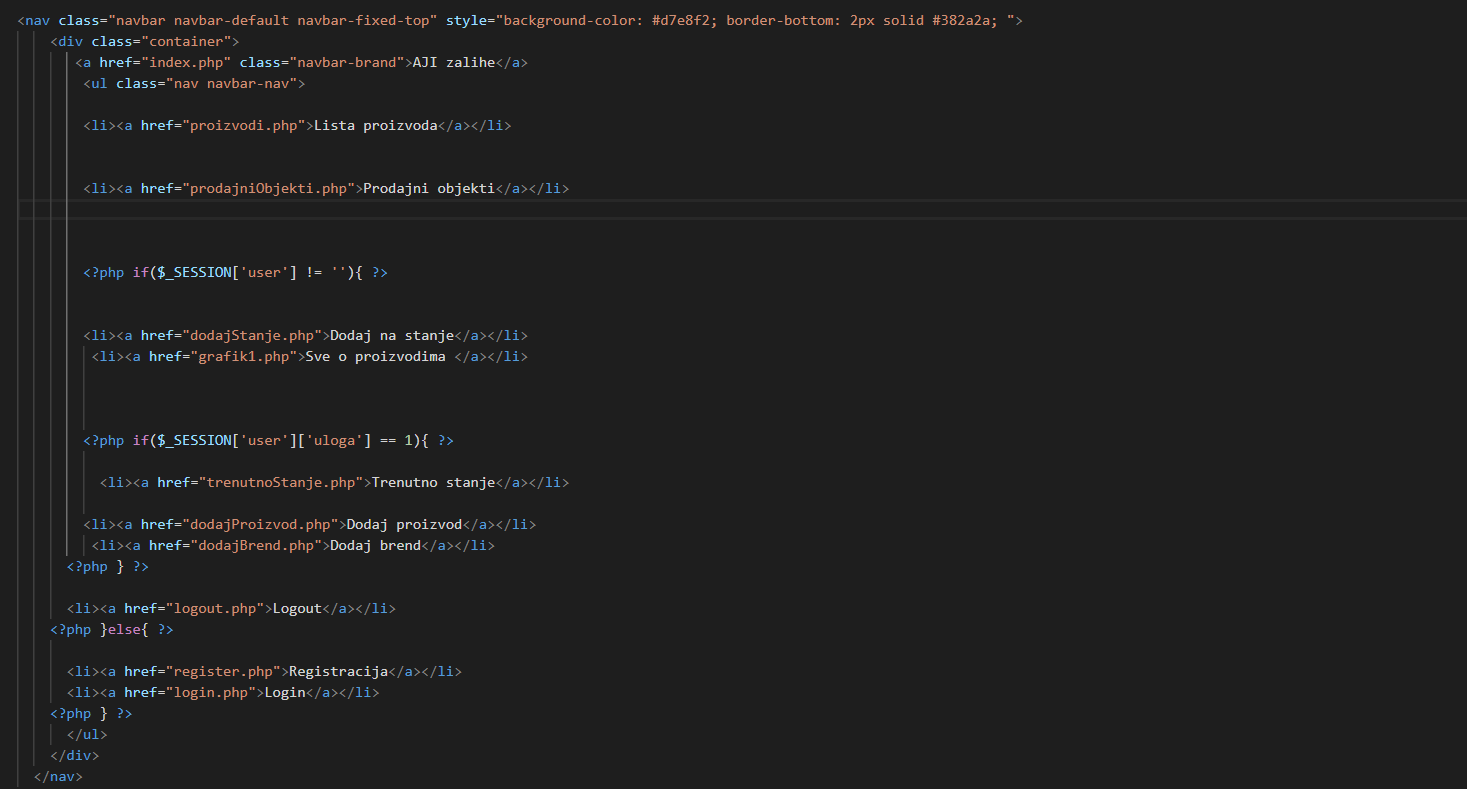
Na sledećih nekoliko slika biće prikazani delovi koda i objašnjene njihove funkcionalnosti.

**Konekcija.php**



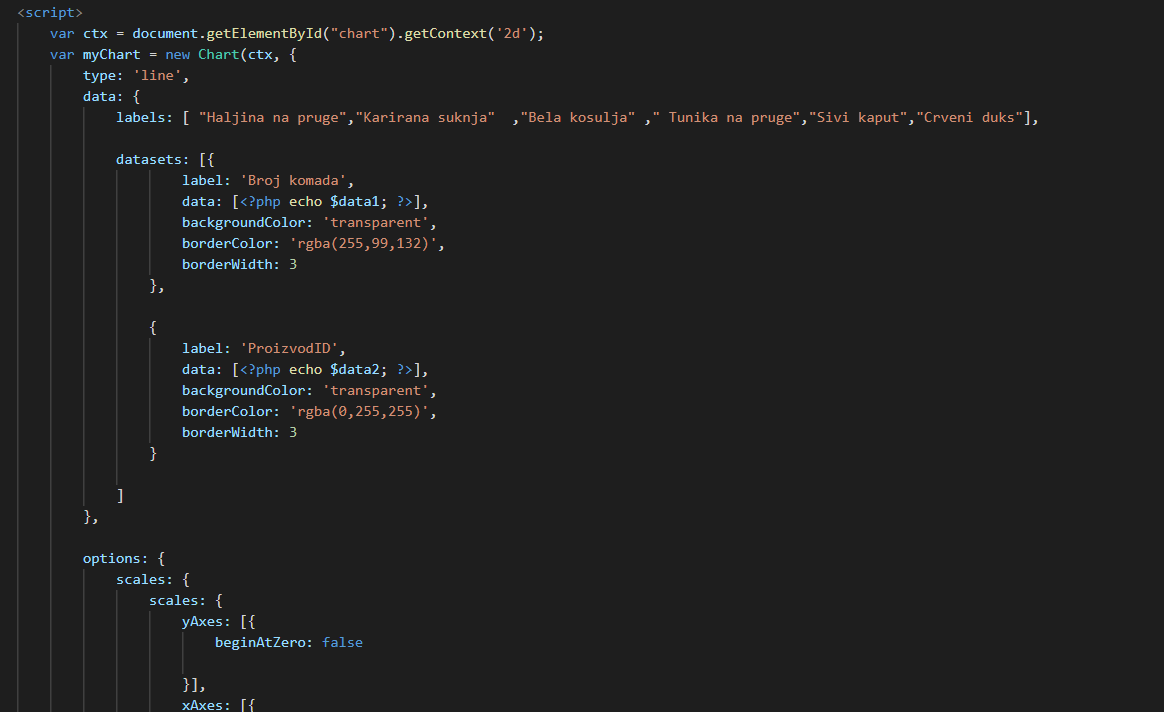
Obzirom da radimo sa bazom podataka neophodno je izvšiti konekciju sa istom.  
Konekciju vršimo tako što instanciramo DBBroker i unesemo odgovarajuće parametre za konekcije pri čemu smo obezbedili upravljanje greškama koje se čuvaju u fajlu „greskeFajl.log“.

**navigacija.php**



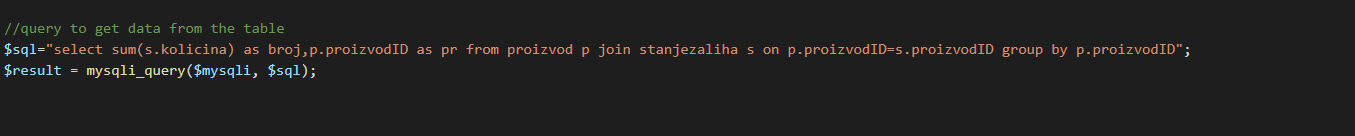
Ovaj deo koda predstavlja formiranje menija. Koristili smo globalnu promenljivu SESSION koja nam daje informaciju da li je ulogovani korisnik običan korisnik ili administrator. U zavisnosti o kom tipu korisnika se radi prikazaće se drugačiji meni kao i pristup različitim funkcionalnostima.

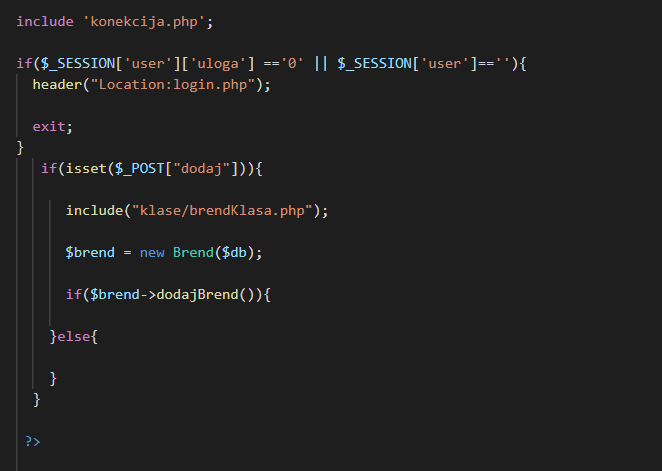
**Podaci o grafiku**

****

Grafik nam prikazuje količinu za svaki proizvid.

Podatke dobijamo spajanjem tabele Proizvod i tabele StanjeZaliha preko spoljnog ključa proizvodID.





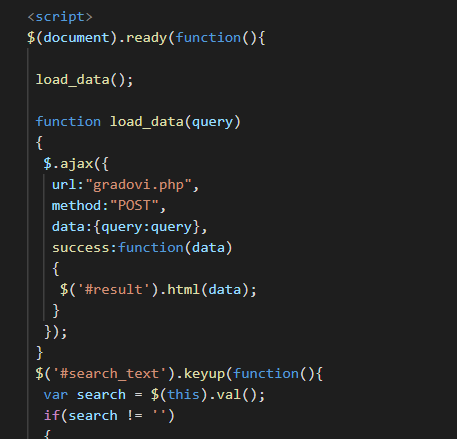
Dodavanje brenda

Posto smo napravili klasu Brend, pravimo njen objekat i pozivamo funkciju dodajBrend, i ubacuje novi brend u bazu podataka.



Prikaz brenova

Pomocu CURL zahteva, uzima json podatke dekodira ih i prikazuje nazive brendova.



Ajax pretraga

Kada hocemo da pretrazimo butike po lokacijama, koristimo ajax pretragu.