Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



**TEMA: Réseau**

**Marketinška agencija**

**Seminarski rad iz Internet tehnologija**

|  |  |
| --- | --- |
|  | Anđelka Čarapić 74/16 |
|  | Ana Cekić 250/16 |

Beograd, 2020.

Sadržaj

[1 Korisnički zahtev 3](#_Toc30150120)

[2 Slučajevi korišćenja 4](#_Toc30150121)

[2.1 Unos usluge 5](#_Toc30150122)

[2.2 Izmena usluge 8](#_Toc30150123)

[2.3 Brisanje usluge 10](#_Toc30150124)

[2.4 Prikaz usluge 13](#_Toc30150125)

[3 Model podataka 16](#_Toc30150126)

[4 Korisničko uputstvo 17](#_Toc30150127)

[4.1 Zajednički slučajevi korišćenja 17](#_Toc30150128)

[4.1.1 Prijava korisnika na sistem 17](#_Toc30150129)

[4.1.2 Odjavljivanje korisnika 17](#_Toc30150130)

[4.1.3 Prikaz svih usluga 18](#_Toc30150131)

[4.2 Administrator 19](#_Toc30150132)

[4.2.1 Rad sa uslugama 20](#_Toc30150133)

[4.2.2 Vizuelizacija 21](#_Toc30150134)

[5 Specifikacija REST servisa 21](#_Toc30150135)

[6 Opis tehnologija korišćenih u aplikaciji 24](#_Toc30150136)

[7 Prikaz reprezentativnih delova koda 26](#_Toc30150137)

[8 GitHub 27](#_Toc30150138)

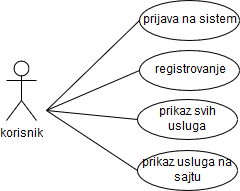
# Korisnički zahtev

Potrebno je projektovati i implementirati veb aplikaciju informacionog sistema marketinske agencije. Sistem treba da radi sa dve vrste korisnika: sa administratorima sistema i korisnicima sistema. Korisnicima sistema treba omogućiti registraciju i logovanje na sistem. Administrator sistema ima pristup istim funkcionalnostima kao i korisnici, a osim tih funkcionalnosti omogućen mu je rad u administraciji.

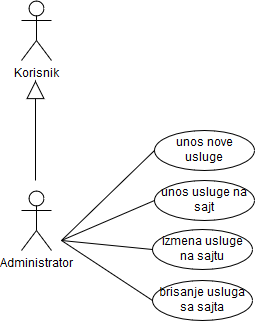
Slučajevi korišćenja:

1. Prijava korisnika na sistem
2. Registrovanje korisnika
3. Unos nove usluge
4. Unos usluge na sajt
5. Izmena usluge na sajtu
6. Brisanje usluge na sajtu
7. Prikaz svih usluga
8. Prikaz usluga na sajtu
9. Odjavljivanje
   1. Odjavljivanje administratora
   2. Odjavljivanje korisnika

# Slučajevi korišćenja

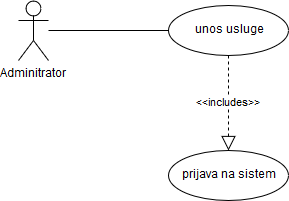


Slika 1: Dijagram slučajeva korišćenja za korisnike



Slika 2: Dijagram slučajeva korišćenja za administratore

## Unos usluge



Slika 3: Slučaj korišćenja – Unos usluge

**Slučaj korišćenja – Unos usluge**

***Naziv SK:* Unos usluge**

***Aktori SK:***Administrator

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

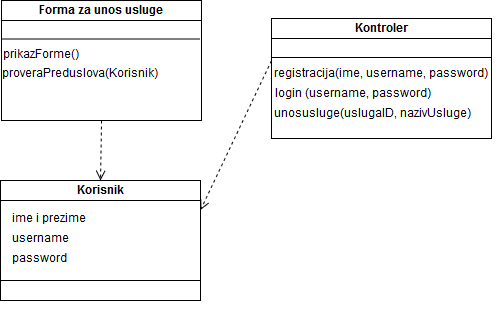
***Preduslov:*** Sistem je pokrenut.

***Osnovni scenario:***

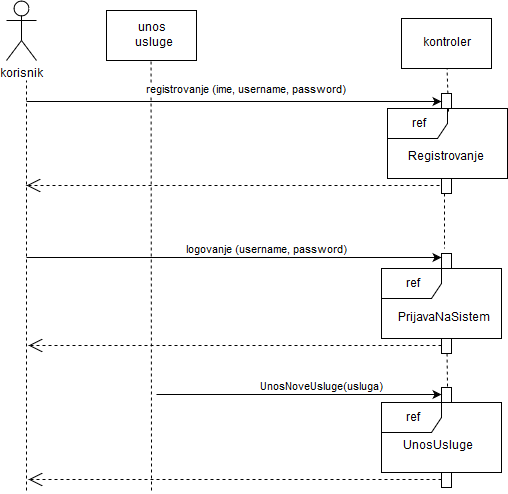
1. Administrator unosi podatke za logovanje.(APUSO)
2. Administrator proverava ispravnost unetih podataka.(ANSO)
3. Administrator poziva sistem da izvrsi logovanje.(APSO)
4. Sistem izvrsava logovanje.(SO)
5. Sistem salje poruku korisniku : ”Uspesno ste se ulogovali”.(IA)
6. Administrator unosi uslugu. (APSO)
7. Sistem čuva uslugu.(SO)

***Alternativni scenario:***

5.1 Ukoliko sistem ne moze da uloguje administratora, salje poruku administratoru: “Niste se ulogovali na sistem.”(IA)

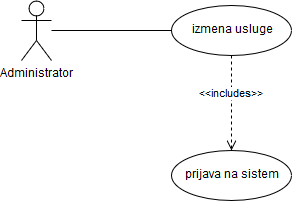


Slika 4: Dijagram klasa –unos usluge



Slika 5: Dijagram sekvenci – Unos usluge

## Izmena usluge



Slika 6: Dijagram slucajeva koriscenja – Izmena usluge

**Slučaj korišćenja – Izmena usluge**

***Naziv SK:*** Izmena usluge

***Aktori SK:*** Administrator

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

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

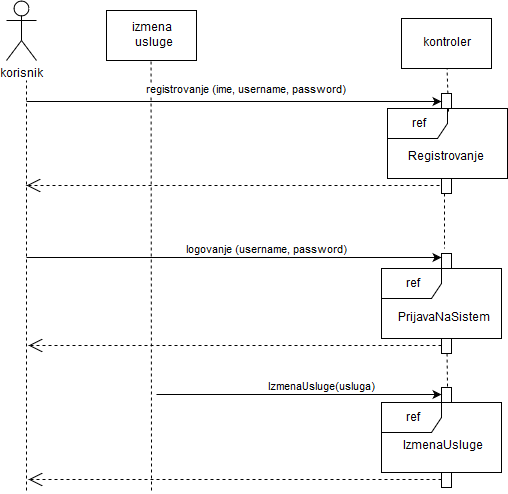
**Osnovni scenario:**

1. Administrator unosi podatke za logovanje.(APUSO)
2. Administrator proverava ispravnost unetih podataka.(ANSO)
3. Administrator poziva sistem da izvrsi logovanje.(APSO)
4. Sistem izvrsava logovanje.(SO)
5. Sistem salje poruku korisniku : ”Uspesno ste se ulogovali”.(IA)
6. Administrator bira uslugu koju želi da izmeni.(APSO)
7. Administrator poziva sistem da zapamti izmenu usluge.(APSO)
8. Sistem pamti izmenu.(SO)
9. Sistem prikazuje adminstratoru poruku:”Uspesno ste izmenili uslugu.”(IA)

**Alternativni scenario**

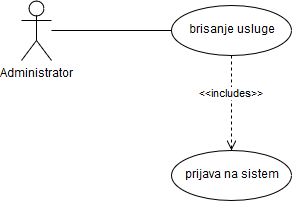
**5.1** Ukoliko sistem ne moze da uloguje administratora, salje poruku administratoru: “Niste se ulogovali na sistem.”(IA)

8.1 Ukoliko sistem ne moze da zapamti izmenu usluge,šalje poruku administratoru: “Niste uspesno izmenili uslugu.”



Slika 7: Dijagram sekvenci–Izmena usluge

## Brisanje usluge



Slika 9: Dijagram slucajeva koriscenja – Brisanje usluge

**Slučaj korišćenja – Brisanje usluge**

***Naziv SK:*** Brisanje usluge

***Aktori SK:*** Administrator

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

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

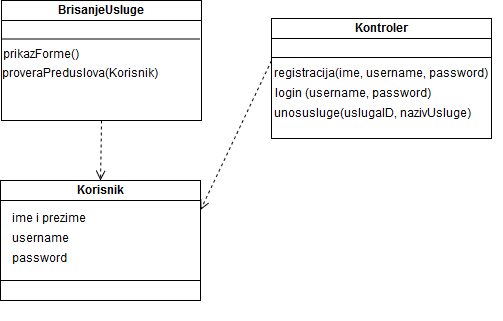
**Osnovni scenario:**

1. Administrator unosi podatke za logovanje. (APUSO)
2. Administrator proverava ispravnost unetih podataka.(ANSO)
3. Administrator poziva sistem da izvrsi logovanje.(APSO)
4. Sistem izvrsava logovanje.(SO)
5. Sistem salje poruku korisniku : ”Uspesno ste se ulogovali”.(IA)
6. Administrator bira uslugu koju treba da obriše.(APSO)
7. Sistem pronalazi u bazi uslugu.(SO)
8. Administrator poziva sistem da obrise uslugu.(ANSO)
9. Sistem prikazuje adminstratoru poruku:”Uspesno ste izbrisali uslugu.”(IA)

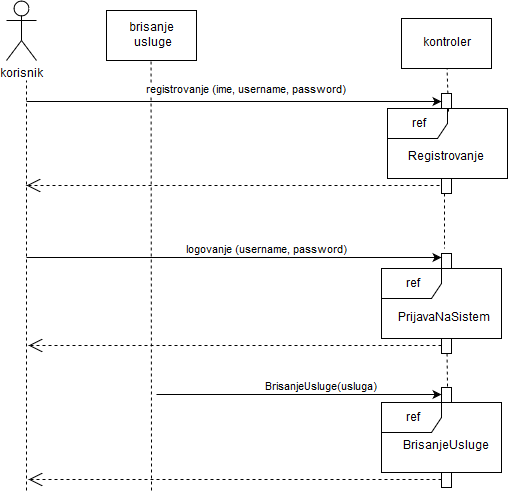
**Alternativni scenario**

5.1Ukoliko sistem ne moze da uloguje administratora, salje poruku administratoru: “Niste se ulogovali na sistem.”(IA)

9.1 Ukoliko sistem ne moze da obrise uslugu , salje administratoru poruku: “Sistem ne moze da obrise uslugu.”

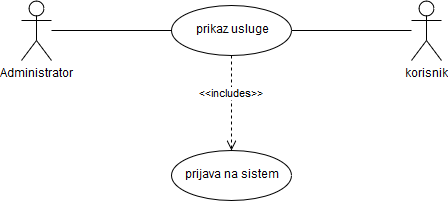


Slika 10: Dijagram klasa–Brisanje usluge



Slika 11: Dijagram sekvenci–Brisanje usluge

## Prikaz usluge



Slika 12: Slučaj korišćenja – Prikaz usluge

**Slučaj korišćenja – Prikaz usluge**

***Naziv SK:* Prikaz usluge**

***Aktori SK:***Administrator, korisnik

***Učesnici SK:***Administrator, korisnik i sistem

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

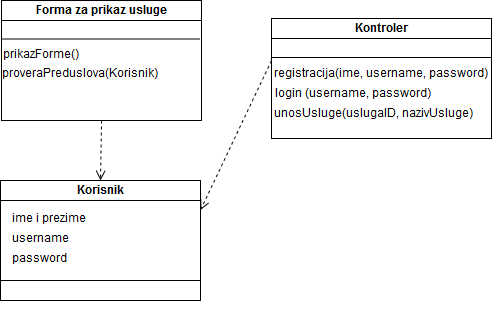
***Osnovni scenario:***

1. Korisnik se registruje na sistem.(APSO)
2. Sistem prihvata registraciju.(SO)
3. Sistem salje poruku korisniku: ”Uspesno ste se registrovali”.(IA)
4. Korisnik salje zahtev sistemu za logovanje.(APSO)
5. Sistem prihvata korisnika. (SO)
6. Sistem salje poruku korisniku : ”Uspesno ste se ulogovali”.(IA)
7. Korisnik bira uslugu koju želi da se prikaze. (APSO)
8. Sistem sistem prikazuje uslugu.(SO)

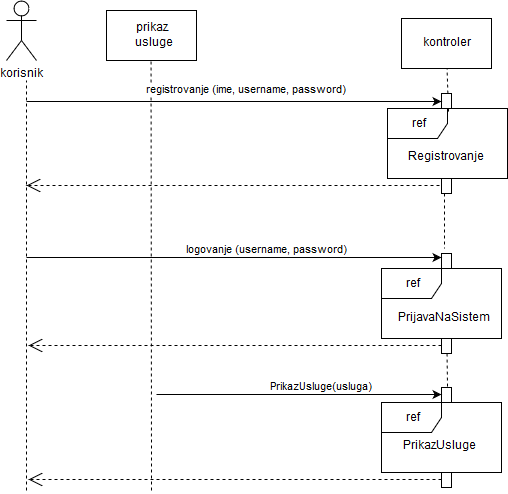
***Alternativni scenario:***

3.1 Ukoliko sistem ne moze da prihvati registraciju, salje poruku korisniku: “Niste se registrovali na sistem.”(IA)

6.1 Ukoliko sistem ne moze da uloguje korisnika, salje poruku korisniku: “Niste se ulogovali na sistem.”(IA)

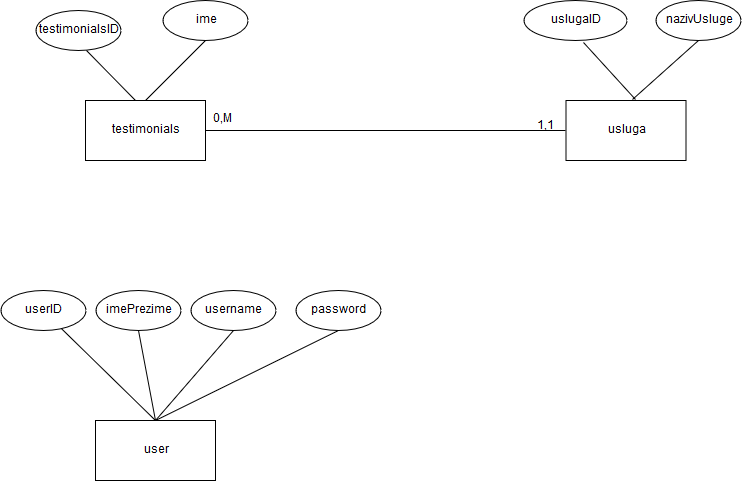


Slika 13: Dijagram klasa –prikaz usluge



Slika 14: Dijagram sekvenci–Prikaz usluge

# Model podataka



0,M

1,1

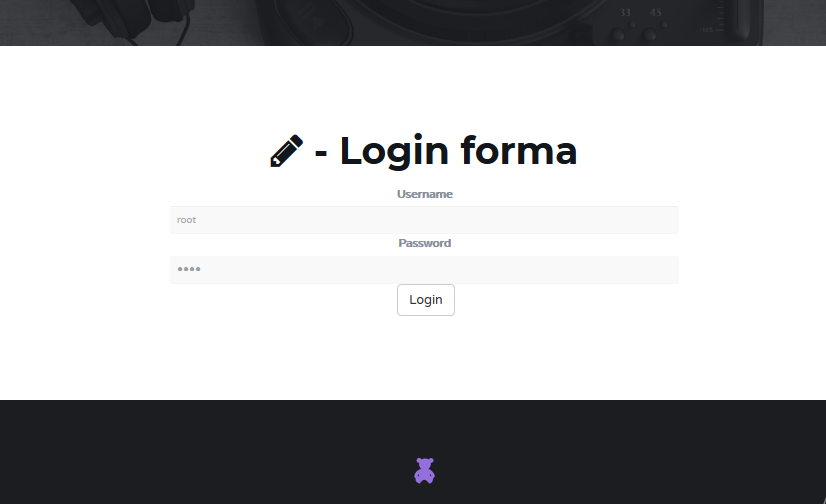
# Korisničko uputstvo

Veb aplikacija marketinske agencije sa  uslugama koje pruza služi da bi zaiteresovani klijenti ovih usluga mogli na brz i jednostavan način da se upoznaju sa istim. Na našoj aplikaciji imamo različite vrste ajax poziva, CRUD operacije, tabelarni prikaz podataka, vizuelizaciju, mapu…

## Zajednički slučajevi korišćenja

### Prijava korisnika na sistem

Prilikom pokretanja aplikacije, potrebno je da se korisnik i administrator uloguju unosom korisnickog imena i lozinke.



### Odjavljivanje korisnika

Na aplikaciji postoji dugme za odjavu korisnika.

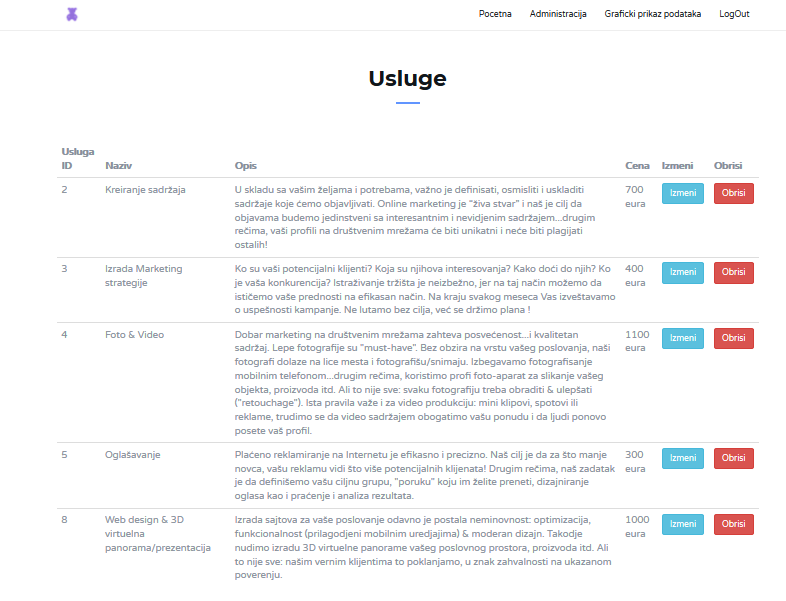
### Prikaz svih usluga

I administrator i korisnik mogu da vide prikaz svih usluga. To su sve usluge koje se trenutno nalaze u nasoj aplikaciji.



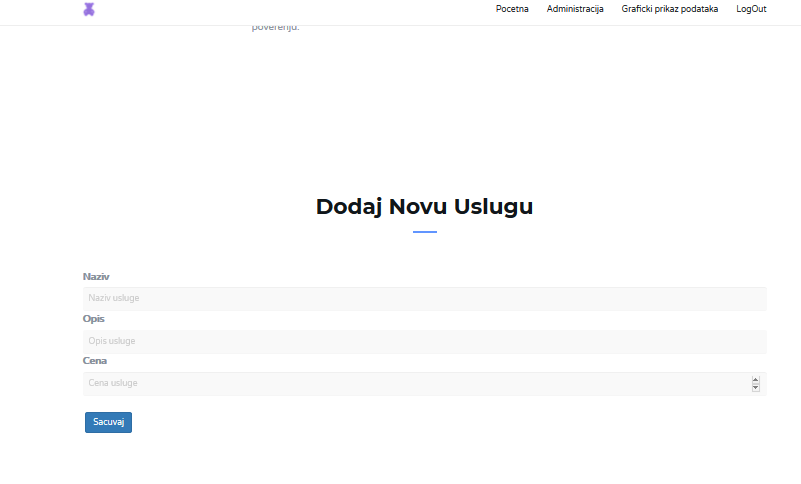
## Administrator

Administrator ima pristup admin delu sajta.



### Rad sa uslugama

Prijavljivanjem administratora na sistem, on ima mogucnost dodavanja usluga, kao I brisanje I izmena istih.



Popunjavanjem polja I klikom na sačuvaj, dodaje se nova usluga u bazu.

### Vizuelizacija

Prijavljivanjem administratora na sistem, on ima mogućnost pregleda podataka u vidu grafika i histograma.



# Specifikacija REST servisa

U dokumentaciji se adrese do pojedinih funkcija veb servisa navode u relativnom obliku, u odnosu na root putanju.

Funkcije servisa:

|  |  |
| --- | --- |
| Opis funkcije | Pregled svih usluga |
| HTTP metoda | GET |
| URL | /usluge.json |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribute uslugaID[int], nazivUsluge[varchar], cena[int]  Primer:  [      {         **"uslugaID"**:"1",       **"nazivUsluge"**:"Kreiranje sadrzaja ",       **"cena"**:"700"    },    {         **" uslugaID "**:"2",       **" nazivUsluge "**:"izrqada marketing strategije",       **"cena"**:"400"     },    {         **" uslugaID "**:"3",       **" nazivUsluge "**:"Foto & video",  **"cena"**:"1100"     },    {         **" uslugaID "**:"4",       **" nazivUsluge "**:"Oglasavanje",       **"cena"**:"300"     } ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Pregled testemonijala |
| HTTP metoda | GET |
| URL | /testimoniali.json |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribute testimonialsID[int], ime[varchar], slika[varchar].  Primer:  [      {         **"** **testimonialsID "**:"1",       **"ime"**:"Milica Avramovic",       **"slika"**:"milica.jpg"     },    {         **"** **testimonialsID "**:"2",       **" ime "**:"Marija Djurovic",       **" slika "**:"marija.jpg"    },    {         **"** **testimonialsID "**:"3",       **" ime "**:"Milos Rokvic",       **" slika "**:"milos.jpg"  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Pregled jedne usluge |
| HTTP metoda | GET |
| URL | /usluge/@ID.json |
| URL parametri | ID |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribute uslugaID[int], nazivUsluge[varchar], cena[int]  Primer:  [      {         **"uslugaID"**:"1",       **"nazivUsluge"**:"Kreiranje sadrzaja ",       **"cena"**:"700"    }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Unos nove usluge |
| HTTP metoda | POST |
| URL | /novaUsluga.json |
| URL parametri | (nema) |
| HTTP body parametri | uslugaID[int], nazivUsluge[varchar],  opis[varchar], cena[int] |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | True/ False |
| Format izlaznih parametara | application/json |

# Opis tehnologija korišćenih u aplikaciji

U radu korišćene su sledeće tehnologije:

1. **PHP** (Hypertext Preprocessor) – PHP je projekat otvorenog koda (open source) i jedan je od najboljih i najpoznatijih server-side rešenja. Server side scripting predstavlja vid izvršavanja web skripti na web-u, a samo mu ime kaže da se izvršava na serveru (suprotno od JavaScript-e koja se izvršava na klijentu (client-side)). Program koji se napiše u PHP-u ne zahteva prevođenje (kompajliranje), nego se interpretira pri svakom izvršavanju. PHP interpretator može raditi po PHP CGI 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 veb-servisa.
2. **CSS** (Cascading Style Sheets) – CSS je tehnologija koja omogućuje razdvajanje strukture stranice od njenog izgleda, što predstavlja osnovni princip ove tehnologije. To konkretno znači da informacije koje se žele prezentovati na stranici treba da budu sadržane u HTML fajlu, a opis izgleda stranice i načina predstavljanja informacije treba da se nalazi u drugom fajlu – css fajlu.
3. **HTML** (HyperText Markup Language) - 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). HTML jezik je sačinjen od običnog teksta i tagova. 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.
4. **Bootstrap** - Open-source JavaScript framework, odnosno kombinacija HTML-a, CSS-a i JavaScript-a, razvijen sa ciljem da omogući i olakša razvoj web formi kao i razvoj naprednih web komponenti. Možemo ga zvati Front-End-Framework. Bootstrap je, dakle, kolekcija razvijenih CSS i JavaScript alata i biblioteka. Isto tako je modularnog tipa, što omogućava njegovu dalju lakšu „nadogranju“ i upotrebu sa različitim modulima koje izrađuju nezavisni developeri.
5. **AJAX** (Asynchronous JavaScript and XML) - 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.
6. **jQuery** (JavaScript Query) - jQuery je posebna vrsta JavaScript biblioteke, razvijene sa namenom da bude nadogradnja osnovnog JavaScript-a. jQuery pojednostavljuje njegovu sintaksu i omogućava bolju interakciju između JavaScript-a i drugih programskih jezika namenjenih razvoju Internet aplikacija.

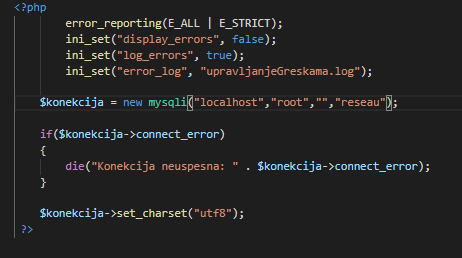
Isto tako, jQuery obezbeđuje jednostavniji pristup DOM-u (Document Object Model), što omogućava kreiranje animacija kao i dinamičkih AJAX web segmenata, daleko jednostavnije nego što bi se to radilo uporebom klasičnog JavaScript-a.

1. **MySQL** (My Standard Query Language) - MySQL je open source sistem za upravljanje relacionim bazama (RDBMS). To znači da su podaci smešteni unutar strukture sposobni da prepoznaju odnose između sačuvanih informacija. Svaka baza podataka sadrži tabele. Svaka tabela (relacija) sadrži jednu ili više kategorija podataka pohranjenih u kolonama. Svaki red sadrži jedinstven podatak (ključ) za kategorije definisane u kolonama.
2. **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.
3. **JavaScript** - JavaScript je objektno zasnovan skriptni jezik. Uključujemo ga u web stranicu da bi je učinili dinamičnijom. 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.

# 7 Prikaz reprezentativnih delova koda

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

**Konekcija.php**

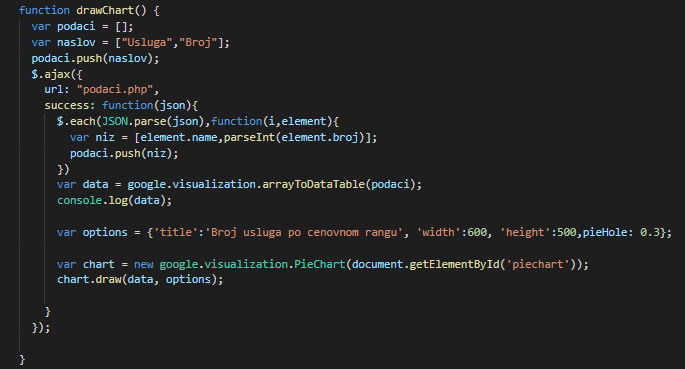


Obzirom da radimo sa bazom podataka neophodno je izvšiti konekciju sa istom.

**Administracija.php**

Primena AJAX-a se nalazi u Administraciji, gde se usluge mogu menjati I brisati.

**Vizuelizacija.php**

****

Formiranje grafiku u obliku pite.

# 8 GitHub

GitHub link ka repozitorijumu: <https://github.com/Andjelka-97/ITEHProjekat.git>