Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



**TEMA: Online prodavnica odeće**

**Seminarski rad iz Internet tehnologija**

Student: Aleksandra Vučičević 126/16

Ivan Umićević 144/16

Jovan Bubonja 1057/18

Beograd, 2020.

Sadržaj

[1. Korisnički zahtev 4](#_Toc30106645)

[2. Opis sistema 5](#_Toc30106646)

[2.1 Opis slučajeva korišćenja 5](#_Toc30106647)

[2.1.1 Prijava na sistem 6](#_Toc30106648)

[2.1.2 Izmena proizvoda 7](#_Toc30106649)

[2.1.3 Brisanje proizvoda 8](#_Toc30106650)

[2.1.4 Dodavanje proizvoda u korpu 9](#_Toc30106651)

[2.1.5 Kreiranje narudžbenice 10](#_Toc30106652)

[2.1.6 Odjava sa sistema 11](#_Toc30106653)

[2.2 Opis arhitekture aplikacije 12](#_Toc30106654)

[2.2.1 Konceptualni dijagram klasa 12](#_Toc30106655)

[2.2.2 Struktura baze podataka 13](#_Toc30106656)

[2.2.3 Struktura fajlova 15](#_Toc30106657)

[2.3. Specifikacija REST API-ja 16](#_Toc30106658)

[2.3.1 Ubacivanje proizvoda u korpu 16](#_Toc30106659)

[2.3.2 Vrati proizvode iz korpe 16](#_Toc30106660)

[2.3.3 Brisanje proizvoda iz korpe 17](#_Toc30106661)

[2.3.4 Pošalji porudžbinu 17](#_Toc30106662)

[3. Opis tehnologija korišćenih u aplikaciji 18](#_Toc30106663)

[3.1 HTML 18](#_Toc30106664)

[3.2 CSS 18](#_Toc30106665)

[3.3 PHP 19](#_Toc30106666)

[3.4 Bootstrap 19](#_Toc30106667)

[3.5 AJAX 20](#_Toc30106668)

[3.6 JSON 20](#_Toc30106669)

[3.7 JavaScript 21](#_Toc30106670)

[3.8 JQuery 21](#_Toc30106671)

[3.9 MySql 22](#_Toc30106672)

[4. Korisničko uputstvo 23](#_Toc30106673)

[5. Prikaz reprezentativnih delova koda 31](#_Toc30106674)

[5.1 Login 31](#_Toc30106675)

[5.2 Logout 31](#_Toc30106676)

[5.3 Primena datatables 32](#_Toc30106677)

[5.4 Vizuelizacija podataka – Google Chart (Javni veb servis) 33](#_Toc30106678)

[5.5 Javni veb servis 34](#_Toc30106679)

[5.6 Listanje proizvoda (JQuery) 34](#_Toc30106680)

[5.7 Količina proizvoda u modalu 35](#_Toc30106681)

[5.8 Dodavanje proizvoda u korpu 35](#_Toc30106682)

[5.9 Korpa 37](#_Toc30106683)

[5.10 Slanje porudžbine 38](#_Toc30106684)

[5.11 Brisanje proizvoda iz korpe 40](#_Toc30106685)

[5.12 Obrada porudžbina 41](#_Toc30106686)

[6. GitHib link ka repozitorijumu 44](#_Toc30106687)

# Korisnički zahtev

U ovom radu biće opisana veb aplikacija tj online prodavnica odeće. Ideja online prodavnice je da ljudi iz udobnosti svog doma mogu da kupuju garderobu. SportPro je naša online prodavnica odeće, jednostavna za koršćenje, na kojoj korisnici mogu da se opuste i prepuste čarima online kupovine.

Pristup veb aplikaciji imaju svi korisnici koji postoje u bazi kao registrovani korisnici. U našem primeru korisnik može biti običan korisnik ili administrator. Omogućena je i odjava korisnika sa sajta, nakon čega se ponovo prikazuje forma za login i korisnik se može ponovo prijaviti.

Korisnik, kao gost prodavnice, može na početnoj strani videti sve dostupne komade odeće koje može poručiti, a zatim može dodavati proizvode u korpu, birati količinu izabranog proizvoda i na kraju kreirati narudžbinu, odnosno kreirati svoju korpu i naručiti odeću na sopstvenu adresu. U bilo kom trenutku, korisnik može i obrisati izabrani proizvod iz korpe. Nakon potvrde kupovine i procesuiranja narudžbine korpa korisika se prazni. Korisnik takođe može videti sve proizvode koji su naručeni, kao i statistiku prodatih proizvoda.

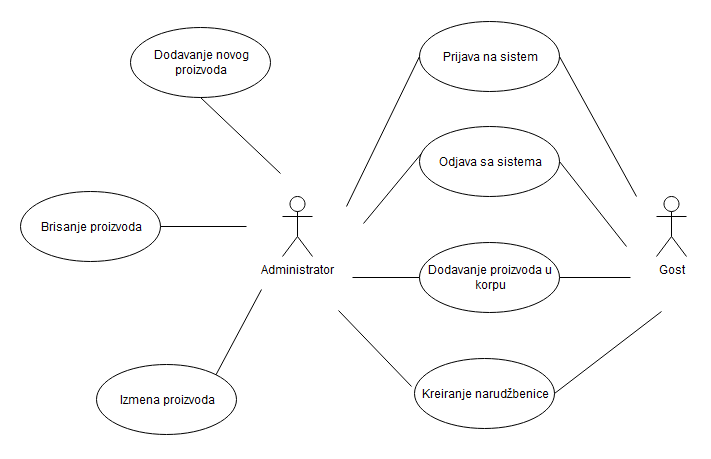
Administrator, u odnosu na korisnika, ima dodatne funkcionalnosti odnosno više opcija. On može dodavati novi proizvod, brisati proizvod ili vršiti izmenu cene proizvoda. Administrator je taj koj obrađuje porudžbine i omogućava da se korisnični zahtev naručivanja garderobe obradi.

# 2. Opis sistema

## 2.1 Opis slučajeva korišćenja

Karakteristični slučajevi korišćenja su:

1. Registrovanje na sistem
2. Prijava na sistem (login)
3. Izmena naloga korisnika
4. Brisanje korisnickog naloga
5. Dodavanje proizvoda u korpu
6. Kreiranje racuna
7. Pretraga prodavnica
8. Konvertovanje valuta
9. Odjava sa sistema (logout)



Slika 1. Dijagram slučajeva korišćenja - korisnik

### 2.1.1 Prijava na sistem

**Naziv SK**

Prijava na sistem

**Aktori SK**

Klijent

**Učesnici SK**

Klijent i sistem (program)

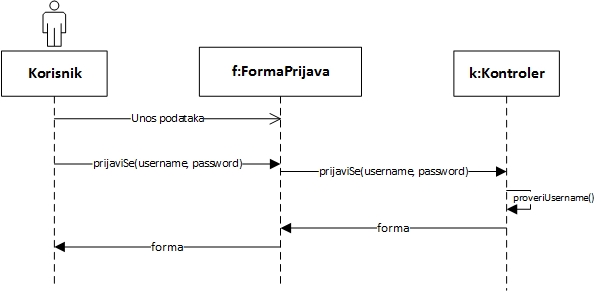
**Preduslov:** Sistem je uključen i prikazuje formu za prijavljivanje klijenta.

**Osnovni scenario SK**

1. Klijent unosi podatke za autentifikaciju klijenta.
2. Klijent poziva sistem da pronađe klijenta sa zadatim podacima.
3. Sistem pretražuje registrovane klijente.
4. Sistem prikazuje početnu formu za prijavljenog klijenta.

**Alternativna scenarija**

* 1. Ukoliko sistem ne može da nađe klijenta prikazuje poruku “Pogresan username I password!”.



### 2.1.2 Izmena proizvoda

**Slučaj korišćenja – Izmena proizvoda**

***Naziv SK:*** Izmena proizvoda

***Aktori SK:*** Administrator

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

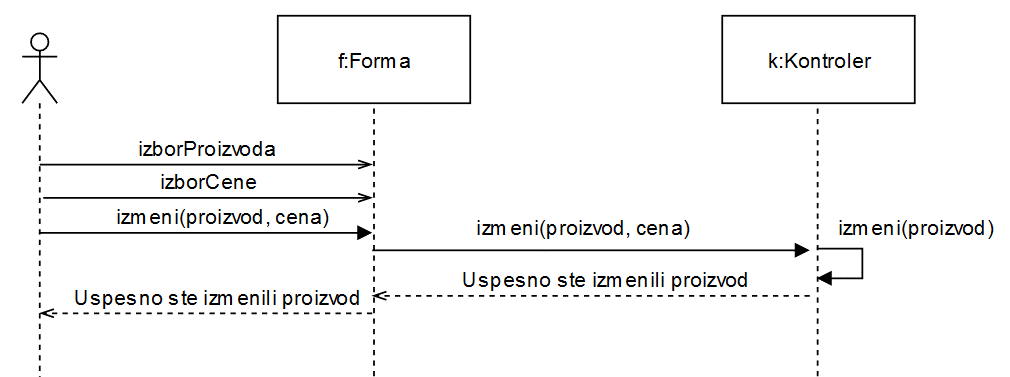
***Preduslov:*** Sistem je pokrenut i korisnik je ulogovan sa svojim podacima.

***Osnovni scenario:***

1. Korisnik otvara aplikaciju i iz menija bira “Izmeni”.
2. Sistem prikazuje formu za izmenu proizvoda.
3. Korisnik unosi podatke koje zeli da izmeni i klikne na dugme “Izmeni”.
4. Sistem prikazuje korisniku da je uspesno izmenjen proizvod i osvežava stranicu.

***Alternativni scenario:***

* 1. Sistem saopštava korisniku da je došlo do greške pri izmeni podataka.



### 2.1.3 Brisanje proizvoda

**Slučaj korišćenja – Brisanje proizvoda**

***Naziv SK:*** Brisanje proizvoda

***Aktori SK:*** Administrator

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

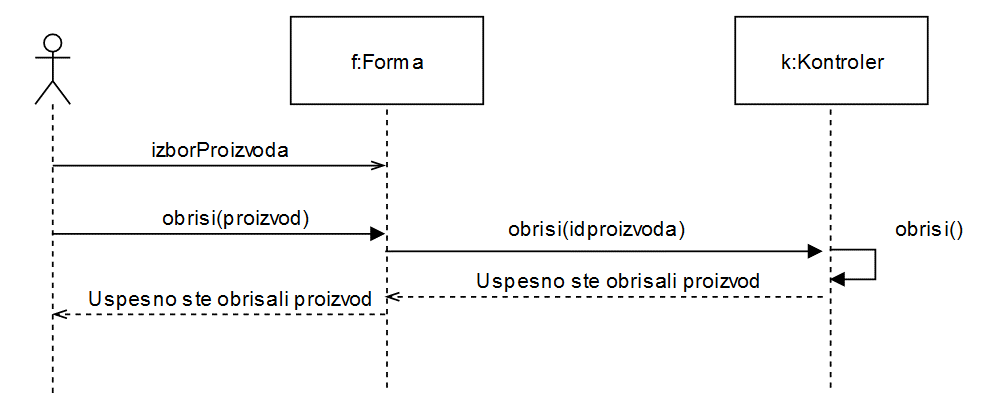
***Preduslov:*** Sistem je pokrenut i korisnik je ulogovan sa svojim podacima.

***Osnovni scenario:***

1. Korisnik otvara aplikaciju i iz menija bira “Obriši”.
2. Sistem prikazuje formu sa popunjenim podacima proizvoda.
3. Korisnik bira proiyvod I pritiska dugme “Obriši”.
4. Sistem prikazuje korisniku da je proiyvod obrisan i osvežava stranicu.

***Alternativni scenario:***

* 1. Sistem saopštava korisniku da je došlo do greške pri brisanju proizvoda.



### 2.1.4 Dodavanje proizvoda u korpu

**Naziv SK**

Dodavanje proizvoda u korpu

**Aktori SK**

Gost, Administrator

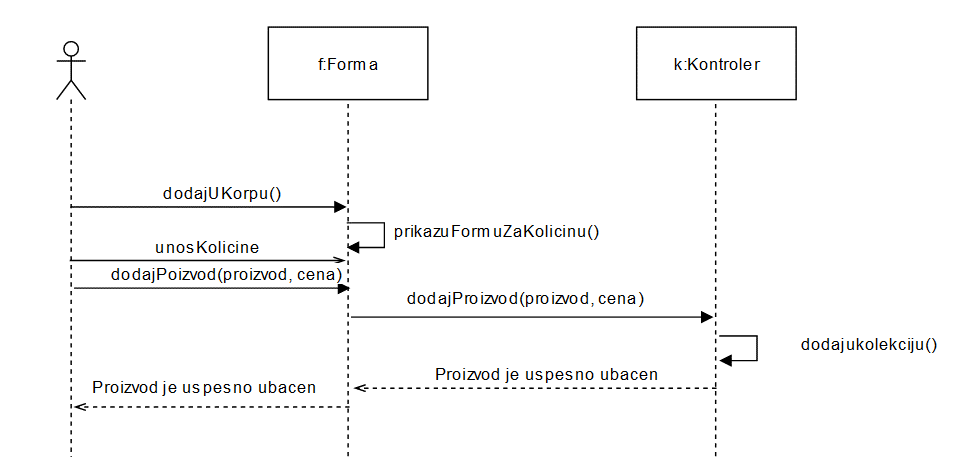
**Učesnici SK**

Gost/Administrator i sistem (program)

**Preduslov**: Sistem je uključen. Sistem prikazuje klijentu formu sa prikazom svih dostupnih proizvoda.

**Osnovni scenario SK**

1. Korisnik poziva sistem da unese izabrani proizvod u korpu.
2. Sistem prikazuje formu za unos količine proizvoda
3. Korisnik unosi količinu proizvoda koju želi.
4. Korisnik pritiska dugme “Dodaj proizvod”.
5. Sistem unosi proizvod u korpu.
6. Sistem prikazuje korisniku poruku “Proizvod je uspešno ubačen”.



**Alternativna scenarija:**

4.1 Ukoliko sistem ne može da unese proizvod u korpu, on prikazuje klijentu poruku: „Proizvod nije ubačen.“

### 2.1.5 Kreiranje narudžbenice

**Naziv SK**

Kreiranje narudžbenice

**Aktori SK**

Administrator/Gost

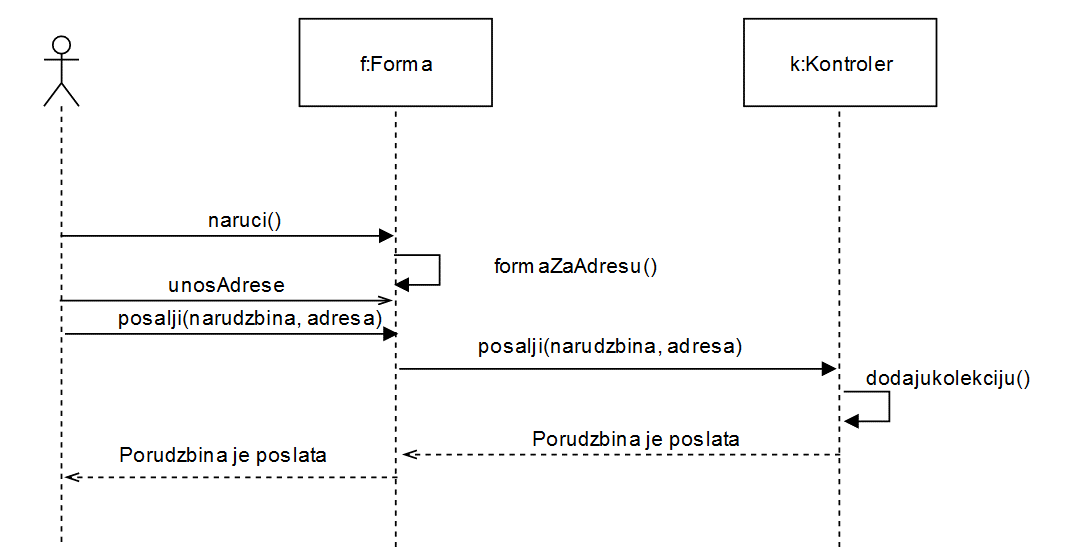
**Učesnici SK**

Administrator/Gost i sistem (program)

**Preduslov**: Sistem je uključen. Sistem prikazuje klijentu formu sa prikazom sadrzaja korpe.

**Osnovni scenario SK**

1. Korisnik poziva sistem da zapamti narudžbenicu.
2. Sistem prikazuje formu za unos adrese
3. Korisnik unosi adresu
4. Korisnik pritiska dugme “Pošalji”
5. Sistem pamti narudžbenicu.
6. Sistem prikazuje poruku “Porudžbina je poslata!”



**Alternativna scenarija:**

* 1. Ukoliko sistem ne može da učita narudžbenicu, on prikazuje klijentu poruku: „Nema proizvoda u korpi“ Prekida se izvršenje scenaria.
  2. Ukoliko sistem ne može da kreira račun, on prikazuje klijentu poruku: „Račun nije kreiran!”

### 2.1.6 Odjava sa sistema

**Naziv SK**

Odjava sa sistema

**Aktori SK**

Administrator/Gost

**Učesnici SK**

Administrator/Gost i sistem (program)

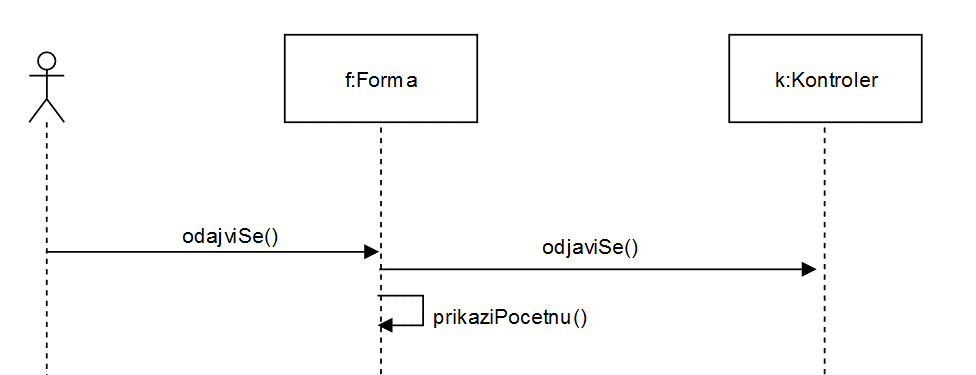
**Preduslov:** Sistem je uključen.

**Osnovni scenario SK**

1. Klijent poziva sistem da ga odjavi.
2. Sistem odjavljuje klijenta.
3. Sistem vraća korisnika na početnu formu za prijavljivanje klijenata.

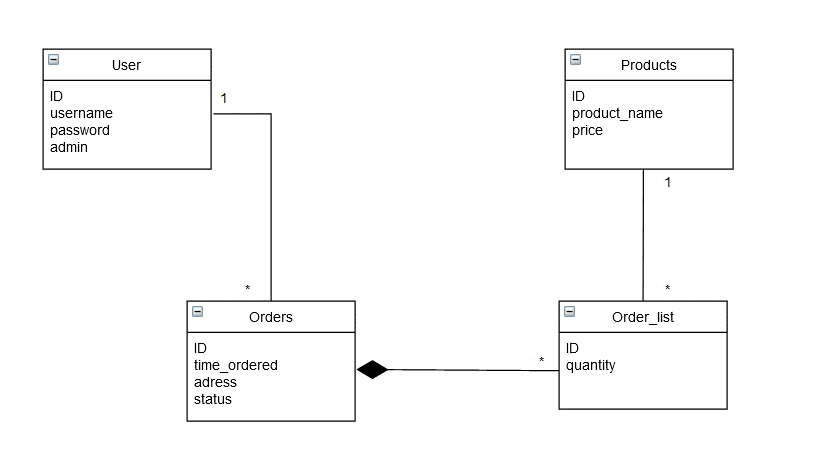
**Alternativna scenarija**

* 1. Ukoliko sistem ne može da odjavi klijenta prikazuje poruku “Došlo je do greške!”.



## 2.2 Opis arhitekture aplikacije

### 2.2.1 Konceptualni dijagram klasa



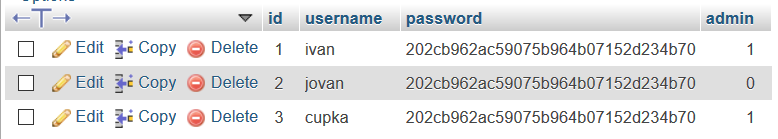
### 2.2.2 Struktura baze podataka

U ovom projektu korišćenja je jedna baza podataka “simple\_shop”.

Baza “simple\_shop” se sastoji iz četiri tabele: User, Product, Orders, Orders\_list. Sve tabele su prikazane na slikama ispod.

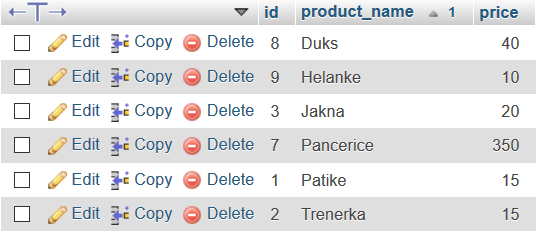
Users

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Lengh/Values |  |
| id | int |  |  |
| username | text |  |  |
| password | text |  |  |
| admin | int |  | 0-nije admin , 1-admin |



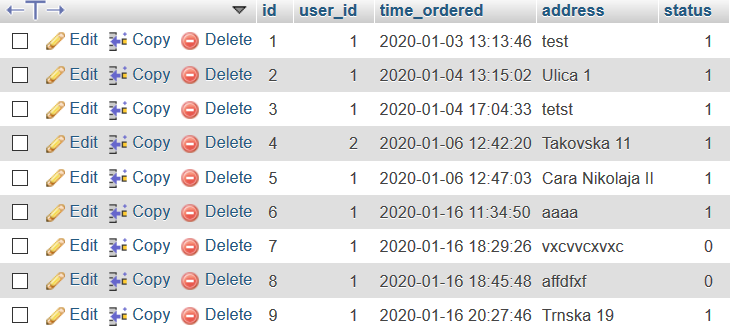
Products

|  |  |  |
| --- | --- | --- |
| Name | Type | Lengh/Values |
| id | int |  |
| product\_name | varchar | 100 |
| price | float |  |



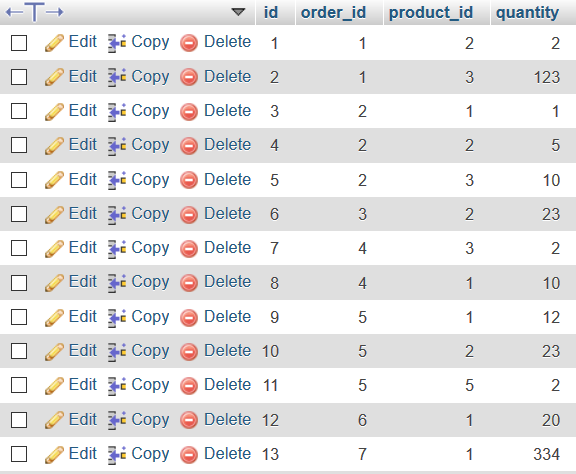
Orders

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Lengh/Values |  |
| id | int |  |  |
| user\_id | int |  |  |
| time\_ordered | datetime |  |  |
| adress | varchar | 100 |  |
| status | int |  | 0-default, 1 = kada je porudžbina obrađena |

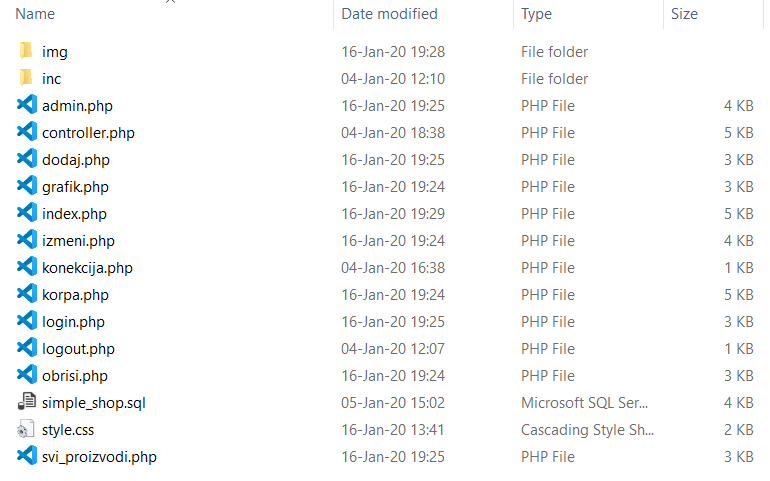


Orders\_list

|  |  |  |
| --- | --- | --- |
| Name | Type | Lengh/Values |
| id | int |  |
| order\_id | int |  |
| product\_id | int |  |
| quantity | int |  |



### 2.2.3 Struktura fajlova





## 2.3. Specifikacija REST API-ja

### 2.3.1 Ubacivanje proizvoda u korpu

|  |  |
| --- | --- |
| Opis funkcije | Ubacivanje proizvoda u korpu |
| HTTP metoda | POST |
| URL | http://localhost:8080/projekat/controller.php?akcija=dodajProizvodUKorpu |
| URL parametri | (nema) |
| HTTP body parametri | {"kolicina":"123","product\_id":"1"}: |
| Format HTTP body parametara | json |
| Izlazni parametri | {"poruka":"Proizvod je uspesno ubacen."} |
| Format izlaznih parametara | json |

### 2.3.2 Vrati proizvode iz korpe

|  |  |
| --- | --- |
| Opis funkcije | Vrati proizvode iz korpe |
| HTTP metoda | GET |
| URL | http://localhost:8080/projekat/controller.php?akcija=vratiProizvodeIzKorpe |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | [{"proizvod\_id":1,"proizvod":"Patike","kolicina":"123", "cena":"15","ukupno":1845}] |
| Format izlaznih parametara | json |

### 2.3.3 Brisanje proizvoda iz korpe

|  |  |
| --- | --- |
| Opis funkcije | Brisanje proizvoda iz korpe |
| HTTP metoda | DELETE |
| URL | http://localhost:8080/projekat/controller.php?akcija= obrisiProizvodIzKorpe&proizvod\_id=3 |
| URL parametri | proizvod\_id: 3 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {"poruka":"Proizvod je uspesno obrisan iz korpe."} |
| Format izlaznih parametara | json |

### 2.3.4 Pošalji porudžbinu

|  |  |
| --- | --- |
| Opis funkcije | Pošalji porudžbinu |
| HTTP metoda | POST |
| URL | http://localhost:8080/projekat/controller.php?akcija=posaljiPorudzbinu |
| URL parametri | (nema) |
| HTTP body parametri | {"adresa":"Trnska 19"}: |
| Format HTTP body parametara | json |
| Izlazni parametri | {"status":1,"poruka":"Porudzbina je poslata!"} |
| Format izlaznih parametara | json |

# 3. Opis tehnologija korišćenih u aplikaciji

U ovom radu korišćene su sledeće tehnologije:

1. HTML (HyperText Markup Language)
2. CSS (Cascading Style Sheets)
3. PHP (Hypertext Preprocessor)
4. Bootstrap
5. AJAX (Asynchronous JavaScript And XML)
6. JSON (JavaScript Object Notation)
7. JavaScript
8. jQuery (JavaScript Query)
9. MySQL (My Standard Query Language)

## 3.1 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 računarski kod su instrukcije koje od računara traže da izvrši određenu operaciju. U osnovi, reč je o posebnoj vrsti sadržaja koji osim konkretnih reči, svakako sadrže i sliku ili video, odnosno audio zapis. Ovaj tip jezika je upravo direktno zadužen za određivanje izgleda svih stranica koje se nalaze u okviru različitih www dokumenata (World Wide Web), te se isto tako primenjuje i u onim situacijama kada je potrebno izvršiti uspostavljanje veze između dokumenata. 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.

## 3.2 CSS

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 prezentirati 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. CSS omogućava definisanje izgleda stranice na jednom mestu kako bi izbegli ponavljanje koda na stranici, jednostavno menjane stranice nakon što je ona kreirana, definisanje veličine fonta, redefinisanje izgleda postojećih tagova u HTML-u, pozicioniranje sadržaja stranice sa prediznošću na nivou piksela i slično.

## 3.3 PHP

PHP je specijalizovani skriptni programski jezik koji se koristi za izradu dinamičkih web stranica, tj. za dinamičko generisanje HTML koda. PHP je open source softver i kompatibilan je sa skoro svim serverima koji se danas koriste (Apache, IIS, itd. ). Nastao 1995. godine napravio Rasmus Lerdorf iz PHP/FI. Koristio je Perl skripte na svojim web stranicama. On je taj program nazvao “Alat za ličnu prezentaciju”. Kasnije je na to dodavao funkcije iz programskog jezika C za komunikaciju s bazama podataka i postavljanja na server dinamičkih web stranica. PHP dozvoljava interakciju sa velikim brojem relacionih baza podataka kao što su MySQL, Oracle, IBM D2, Microsoft SQL Server, PostgreSQL i SQLite. Koristi se za upravljanje dinamičkim sadržajem, praćenjem sesija, zapravo PHP vam omogućava da uz pomoć njega napravite čitave e-commerce sajtove. 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.

## 3.4 Bootstrap

Bootstrap je alat za brzi razvoj web aplikacija. To je kolekcija CSS i HTML pravila i JavaScript ekstenzija koja koristi neke najnovije tehnike pretraživača, pruža modernu tipografiju, forme, dugmad, tabele, okvire i navigaciju itd. Na samim počecima Twitter-a, inženjeri su koristili skoro svaku biblioteku kako bi se upoznali sa front-end zahtevima. Bootstrap je razvijen tokom prvog Twitter Hackweek-a, kada je objavljena prva stabilna verzija koju su inženjeri mogli koristiti u kompaniji. Uz pomoć povratne infomacije od mnogih inženjera, Bootstrap je značajno porastao i obuhvatio je ne samo osnovne stilove, već i elegantne i trajne front-end obrasce. Prva verzija Bootstrap-a objavljena je u avgustu 2011. godine kao open source izdanje, a u februaru 2012. je postao najpopularniji projekat na GitHub-u. U svom jezgru Bootstrap je CSS, ali je izgrađen uz pomoć Less-a, fleksibilnog preprocesora koji pruža mnogo više mogućnosti i fleksibilnosti za korisnike nego obični CSS. Sa Less-om dobija niz funkcija kao što su ugnježdene deklaracije, varijable, operatori i još mnogo opcija. Pored toga pošto je Bootstrap čisti CSS, kada se kompajlira putem Less-a, dobijaju se dve važne prednosti: Prvo, Bootstrap se veoma lako inplementira, samo je potrebno ubaciti ga u kod. Kompaliranje Less-a se može postići putem JavaScript-a, pomoću Mac aplikacije ili preko Node.js. Drugo, kada se jednom kompajlira Bootstrap sadrži samo CSS, što znači da nema suvišnih slika, Flash-a ili Java Script-a.

## 3.5 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.

## 3.6 JSON

JSON 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. Prva je zbirka parova ime / vrednost. Na raznim jezicima, to je realizovano kao objekat, zapis, struktura, rečnik, heš tabela, lista sa ključevima ili asocijativni niz. Druga je 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.

## 3.7 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. JavaScript jezik je nastao u kompaniji Netscape pod prvobitnim imenom LiveScript koje mu je dodelio direktor projekta Brendan Eich. Početkom decembra 1995. godine, Netscape i Sun su zajednički objavili da će se skript jezik ubuduće zvati JavaScript. Iako je Netscape imao nekoliko dobrih marketinških razloga za usvajanje ovog imena, izmena je možda izazvala više konfuzije nego što je iko očekivao. Ispostavilo se da je prava poteškoća bila da se svetu objasni razlika između jezika Java i jezika JavaScript. JavaScript je objektno zasnovan skriptni jezik. Objektno je zasnovan jer programer ne definiše samo tip podataka, nego i vrstu operacija (funkcija) koje se mogu primeniti 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 primjer, objekti mogu da pridobiju karakteristike od drugih objekata. Skriptni je jezik jer se sastoji od serije komandi koje se očitavaju u interpreteru, a da se predhodno ne kompajlira sadržaj. Odnosno, ne prevodi se u mašinski jezik, već se komande direktno "čitaju" iz koda. Zbog ove karakteristike JavaScript se izvršava na strani korisnika, tj. na računaru na kojem je pokrenut sadržaj sa JavaScript-om. Uključujemo ga u web stranicu da bi je učinili dinamičnijom. 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).

## 3.8 JQuery

jQuery je JavaScript biblioteka otvorenog koda koja pojednostavljuje interakcije između HTML dokumenta ili, preciznije, objektnog modela dokumenta (DOM) i JavaScripta. Filozofija biblioteke jQuery je „Piši manje, uradi više“ (engl. write less, do more). Osnovna uloga biblioteke se odnosi na olakšavanje kretanja kroz dokument, selektovanje DOM elemenata, kreiranje animacija, upravljanje događajima i razvoj veb aplikacija. jQuery je moćan alat za kreiranje sofitsticiranih efekata na veb stranici. Istovremeno, jQuery pruža mogućnost razvoja plaginova na osnovu dostupnih funkcionalnosti bibiloteke. Na ovaj način se stvaraju moćne i dinamične veb stranice.

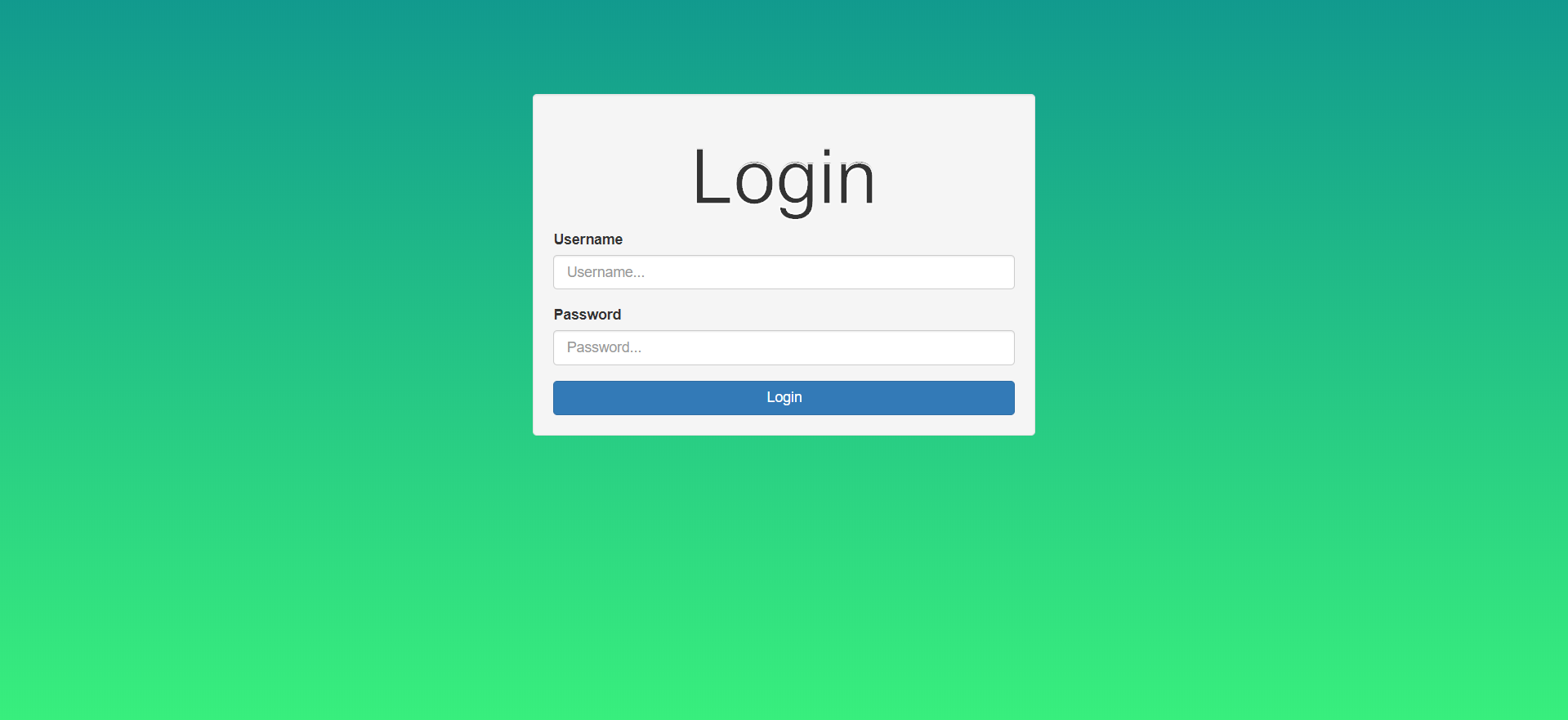
## 3.9 MySql

MySQL je besplatan, open source kod za upravljanje bazom podataka. Ranije u svom razvoju, MySQL baza podataka suočila se s raznim protivnicima jer su joj nedostajale neke osnovne funcije definisane SQL standardom. Naime, MySQL baza je optimizirana kako bi bila brža, na štetu funkcionalnosti. Nasuprot tome, vrlo je stabilna i ima dobro dokumentovane modele i ekstenzije te podršku od brojnih programskih jezika: PHP, Java , Perl, Python.. MySQL baze su relacijskog tipa, koji se pokazao kao najbolji način skladištenja i pretraživanja velikih količina podataka i u suštini predstavljaju osnovu svakog informacijskog sustava, tj. temelj svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija. Kao alat za jednostavno upravljanje MySQL bazom podataka, često se koristi aplikacija PhpMyAdmin. U pitanju je opensource veb aplikacija napisana u programskom jeziku PHP, koja omogućava kreiranje novih baza podataka, izmenu postojećih baza, rad sa tabelama, pregled, unos, izmenu i brisanje podataka, kao i izvršavanje SQL upita.

# Korisničko uputstvo

U ovom odeljku seminarskog rada treba kreirati korisničko uputstvo za sve korisničke uloge, sa pratećim screenshot-ovima. Korisnička uputstva treba kreirati prema slučajevima korišćenja.

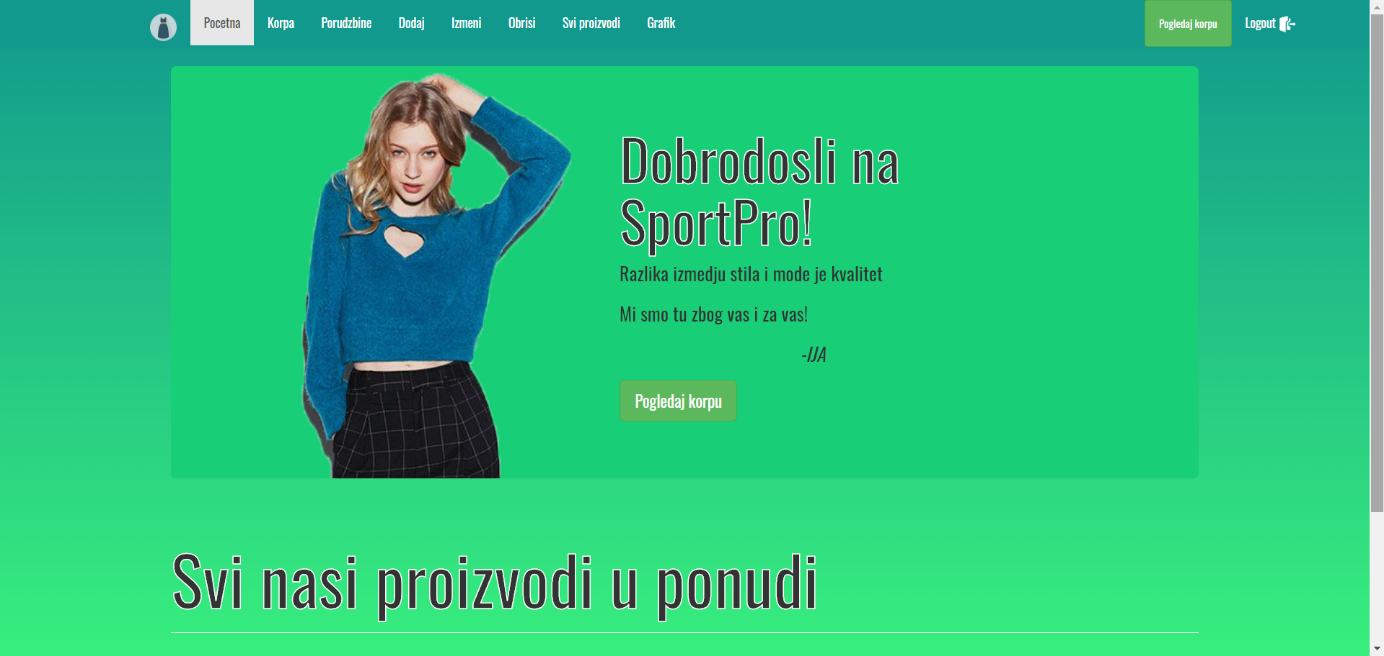
Pri korak na našem sajtu je logovanje.

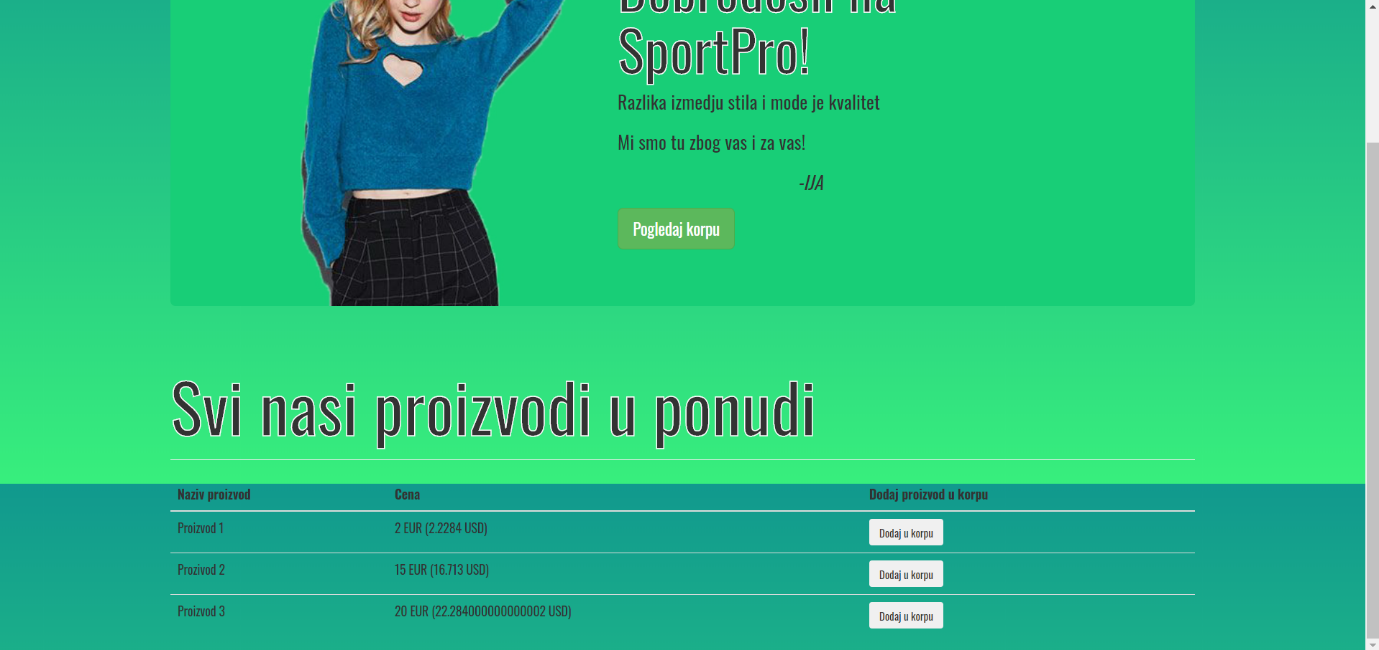


Korisnik se kroz sajt kreće pomoću navigacionog bara.



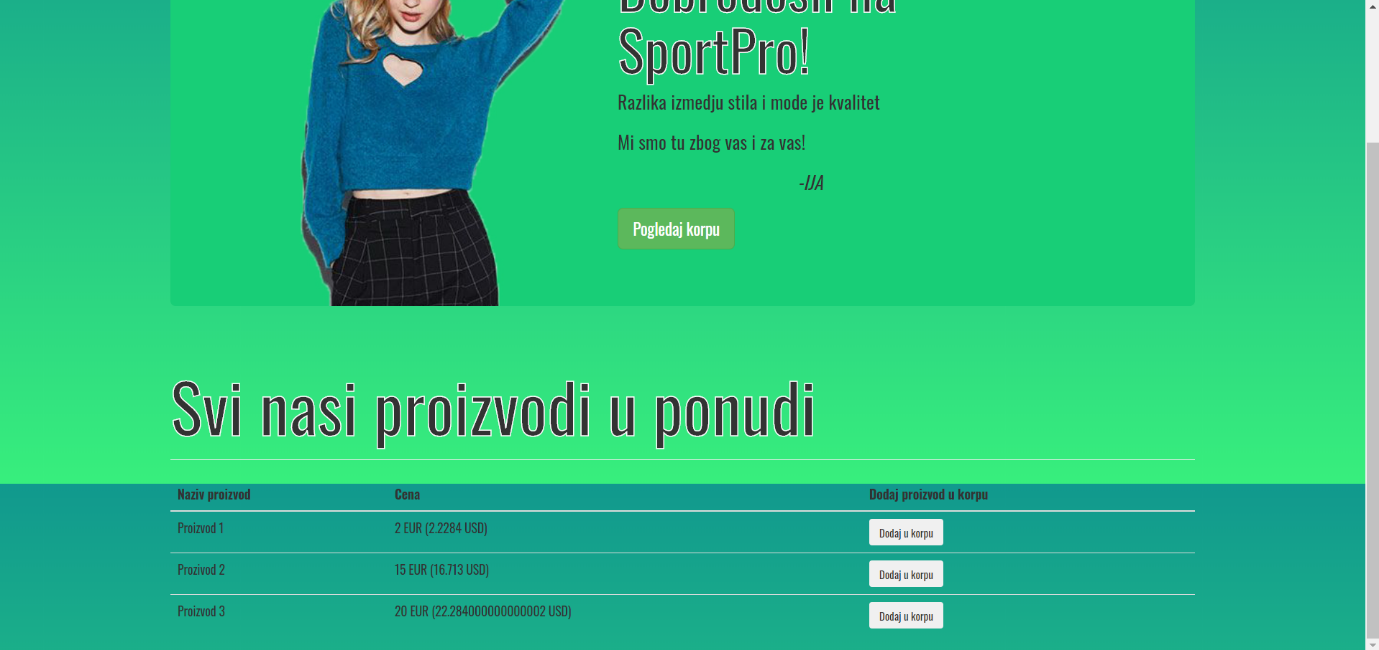
Ukoliko je ulogovani korisnik običan korisnik njemu se prikazuje Početna stranica.



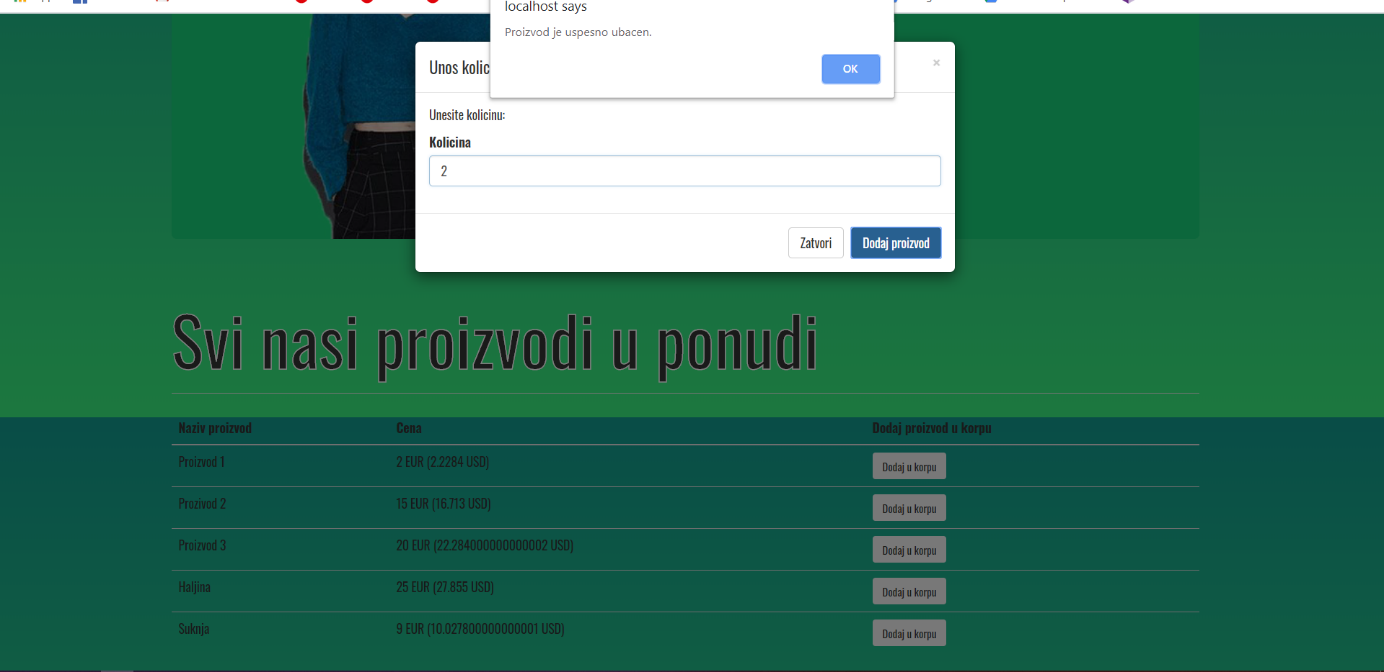


Na početnoj stranici korisnik dobija poruku „Dobro došli na SportPro!”

Na početnoj stranici korisnik kupuje proizvode. Postoji lista svih proizvoda u ponudi. Korisnik može videti naziv kao i cenu proizvoda i klikom na dugme „Dodaj u korpu” da doda u korpu željeni proizvod.

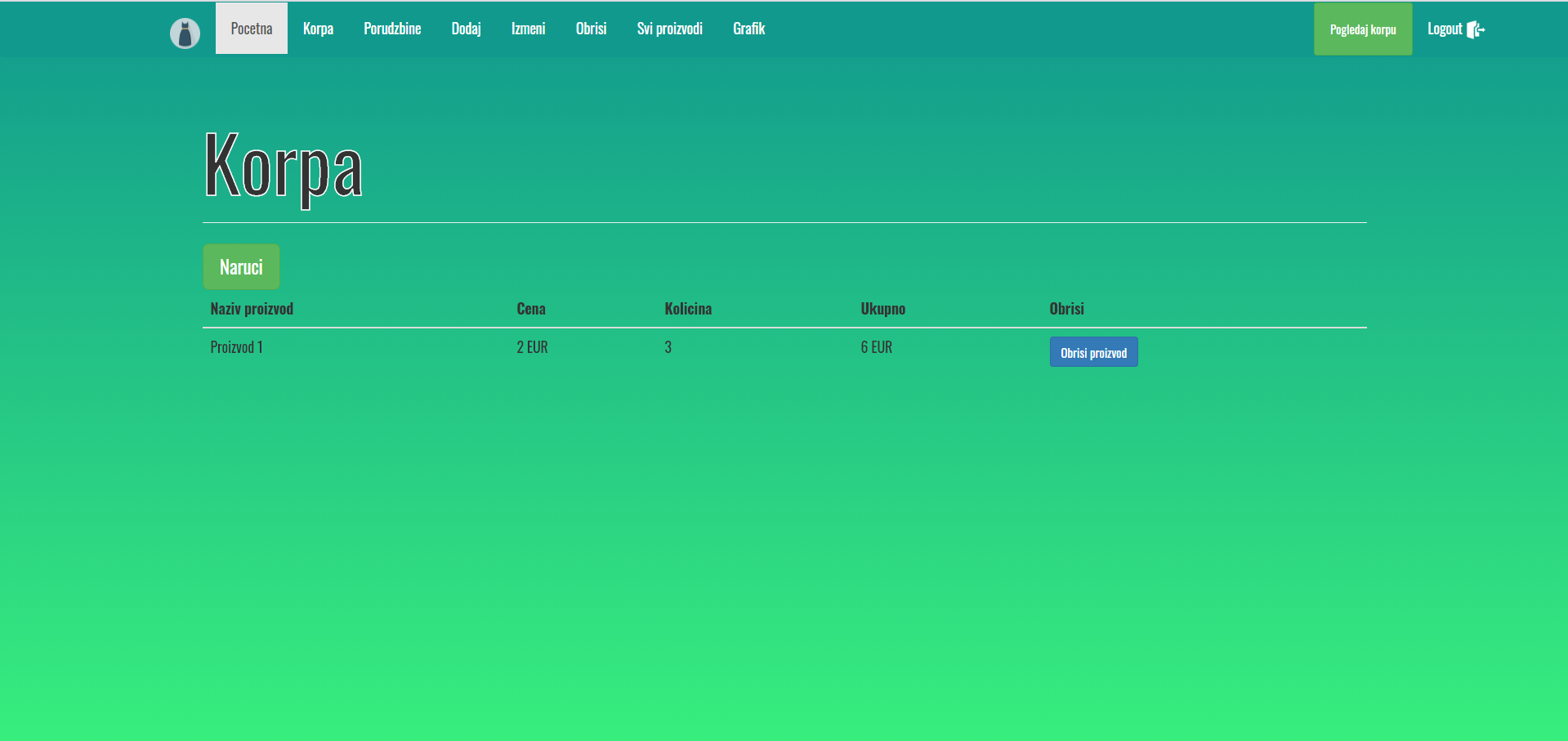


Kada korisnim ubaci u korpu željeni proizvod prikazje mu se poruka.



Klikom na dugme „Pogledaj korpu” korisnik može da pogleda koje sve proizvode je dodao u korpu.

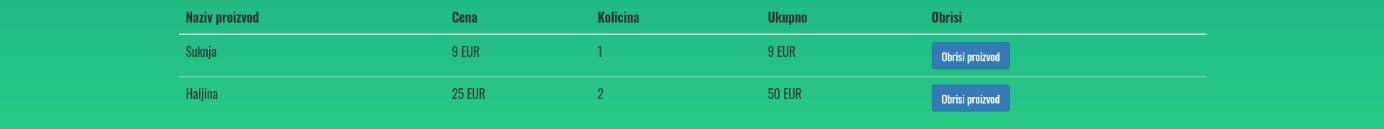
Istu funkcionalnosti imamo i u navigacionom baru, te klikom na „Korpa“ korisnik pristupa svojoj korpi.



Takođe, klikom u desni gornji ugao „Pregledaj korpu” korsnik može videti šta se sve nalazi trenutku u njegovoj korpi.



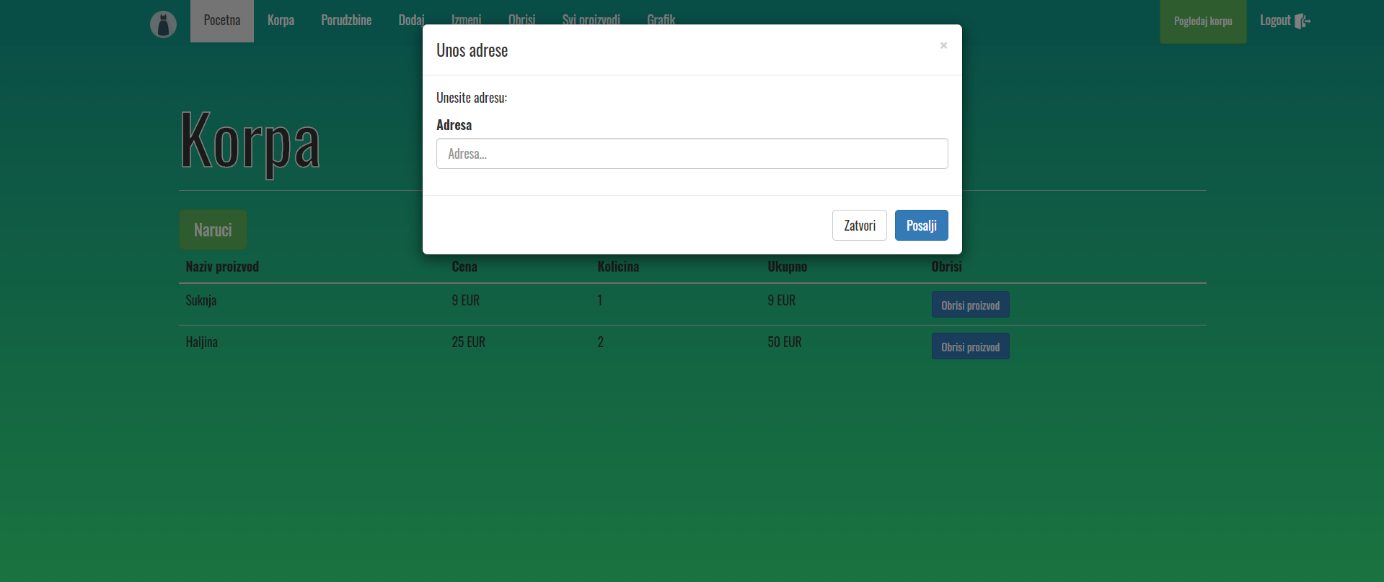
U svakom trenutku u korpi, korisnik ima mogućnost da obriše izabrani proizvod iz korpe.



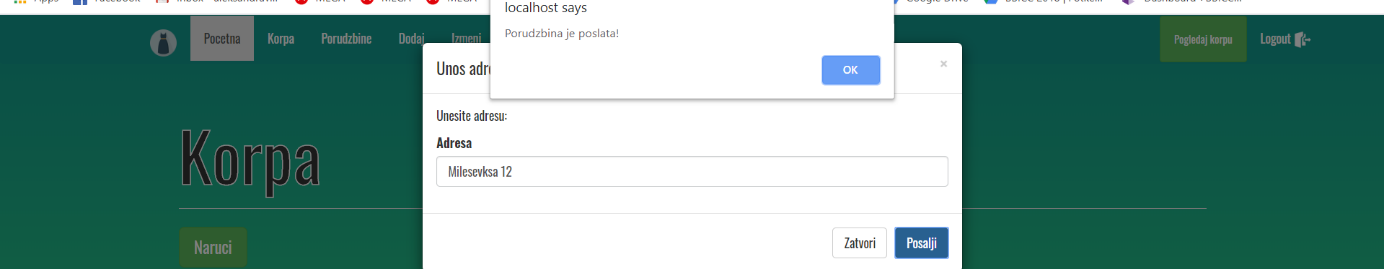
Kada korisnik završi kupovinu on u okviru korpe klikće na dugme „naruči“ i time naručuje proizvode koje je izabrao i koji se nalaze u korpi.



Kada korisnik klikne da naruči proizvode pojavljuje mi se polje u kome je potrebno da unese adresu na koju želi da mu proizvodi budu dostavljeni.



Kada korisnik unese adresu i klikne dugme Pošalji dobija poruku da je porudžbina uspešno poslata i njegova korpa ostaje prazna.



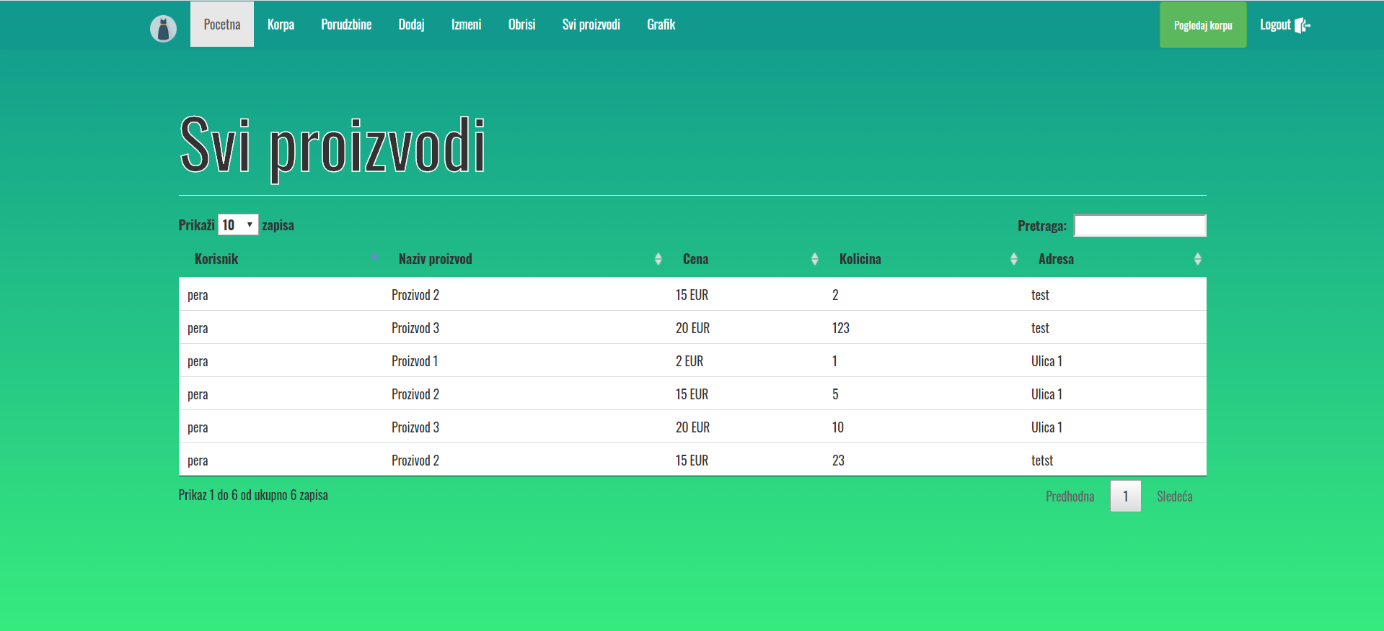
Ukoliko korisnik klikne na ikonicu haljine, biće odveden na Početnu stranicu.

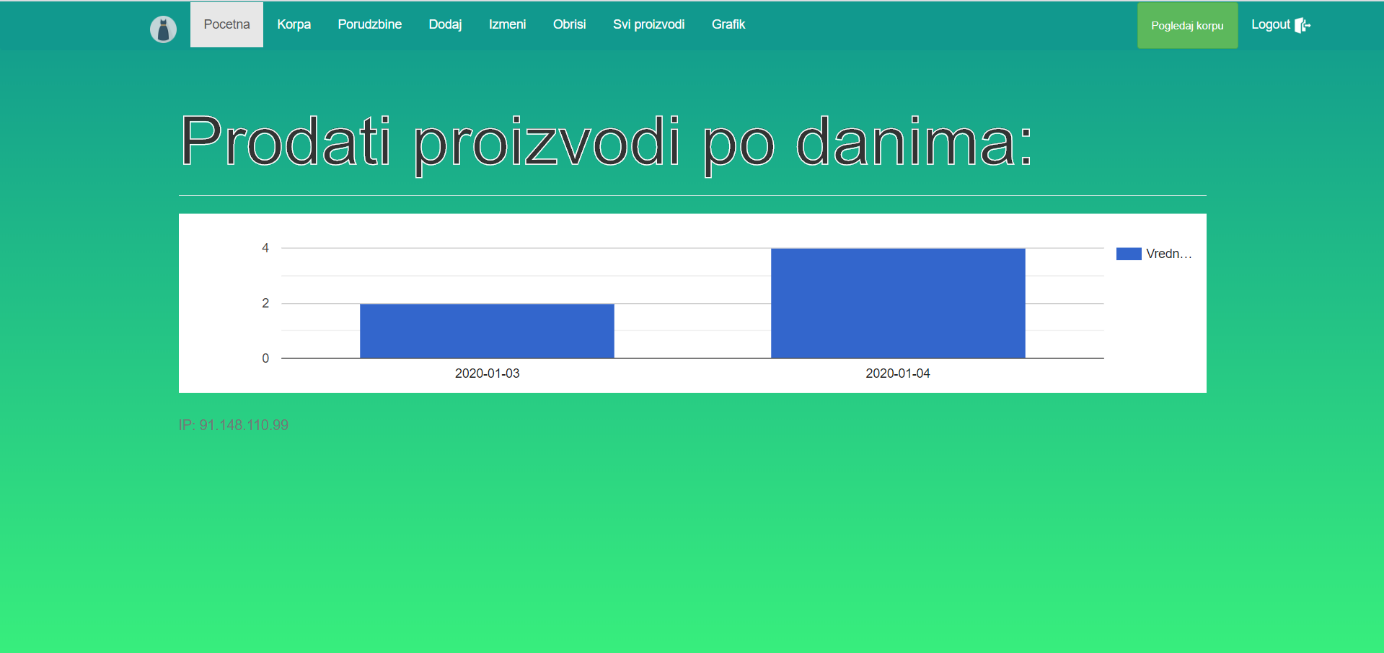


„Porudžbine“, „dodaj“, „izmeni“ i „obriši“ su zabranjeni za korisnika, odnosno prikazaće se Početna stranica, ukoliko je ulogovani korisnik običan korisnik (nema ulogu admina).



Klikom na „Svi proizvodi“ korisnik je kroz preglednu tabelu u mogućnosti da vidi sve proizvode koje je naručivao, kao i da ih pretraži.

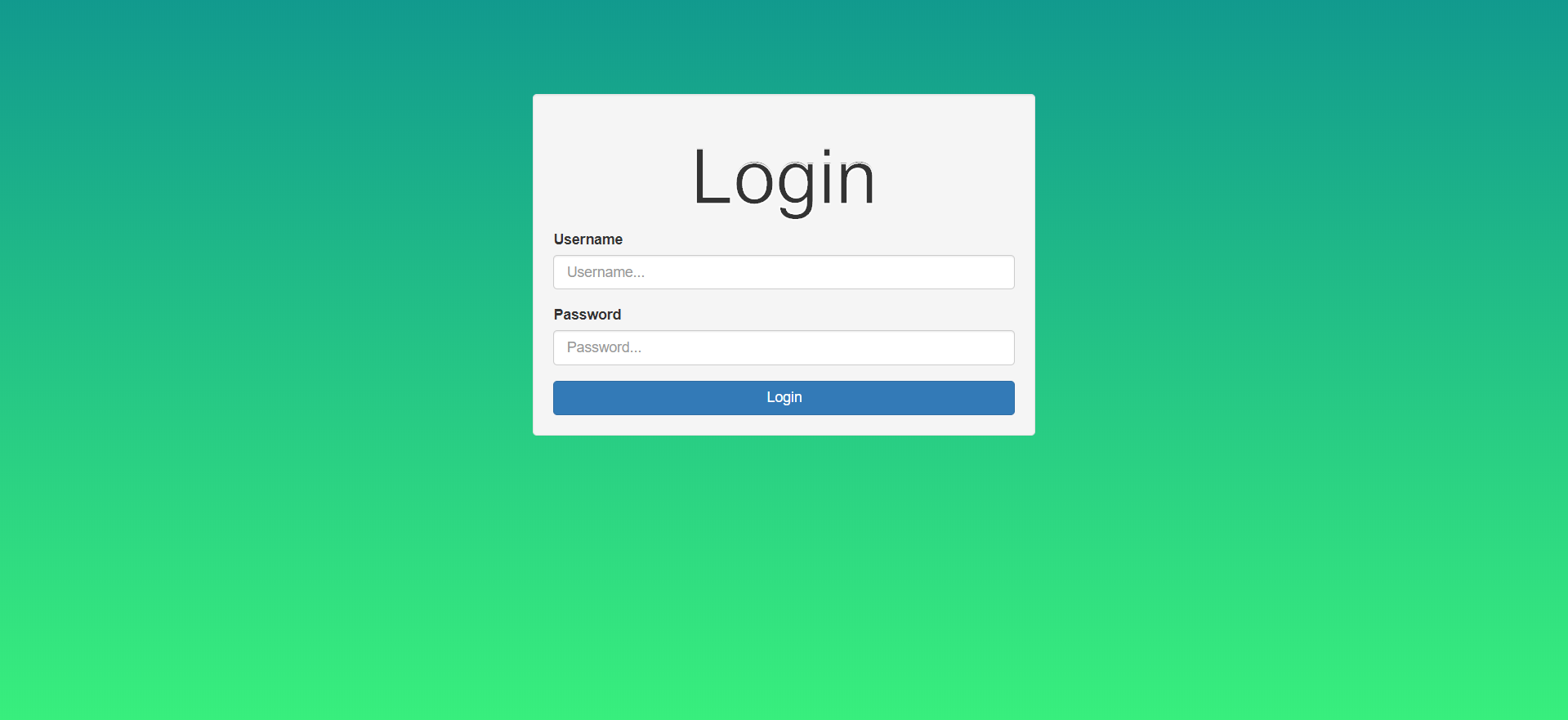


Klikom na „Grafik“ korisnik može videti statistiku prodatih proizvoda po danima. Takođe, korisniku se ispisuje IP adresa sa koje pristupa sajtu u donjem delu ekrana. 

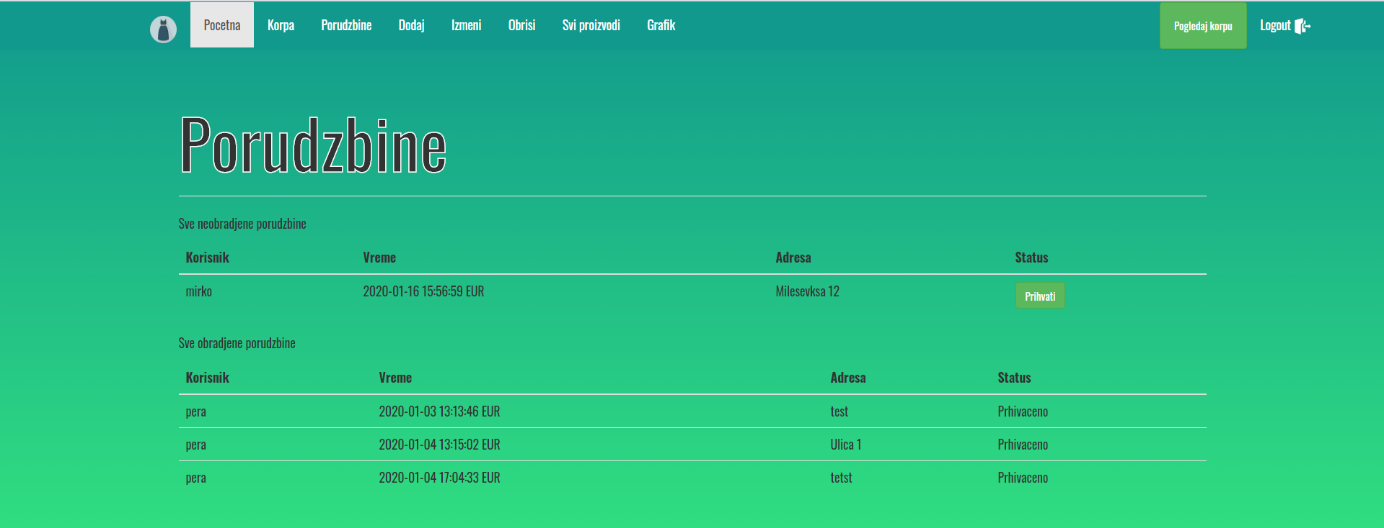
Kada god poželi, korisnik se može izlogovati sa sajta klikom na dugme „Logout“.



Kada se korisnik izloguje ponovo se prikazuje forma za logovanje.



Sada ćemo prikazati slučaj ukoliko je ulogovani korisnik admin. Kada se uloguje, adminu se prvo prikazuje stranica „Porudžbine“.



Admin može da koristi sve funkcionalnosti koje koristi i korisnik, ali i neke dodatne. U nastavku ćemo upravo staviti fokus na njih.

U okviru porudžbina, postoji 2 odaljka. Prvi je „Sve neobrađene porudžbine“. Ove porudžbine su na „čekanju“. Da bi admin prihvatio porudžbinu, neophodno je da klikne na dugme prihvati.



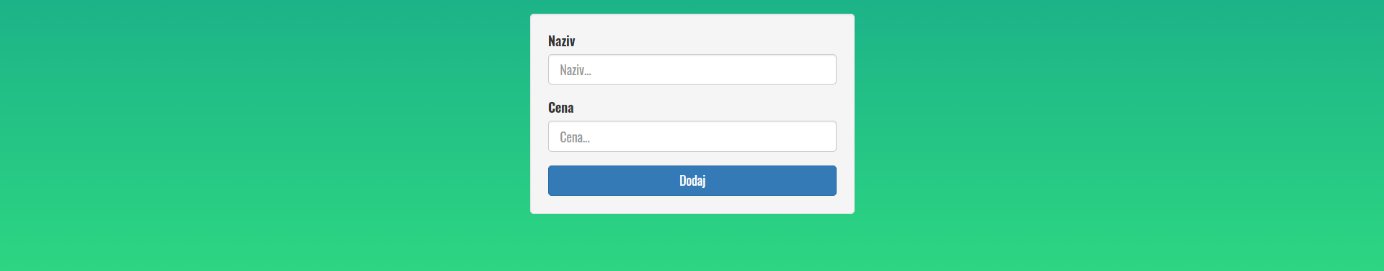
Drugi odeljak je „Sve obrađene porudžbine“. Kada admin prihvati porudžbinu, porudžbina prelazi u odeljak sve obrađene porudžbine, gde se nalaze sve prihvaćene porudžbine korisnika.



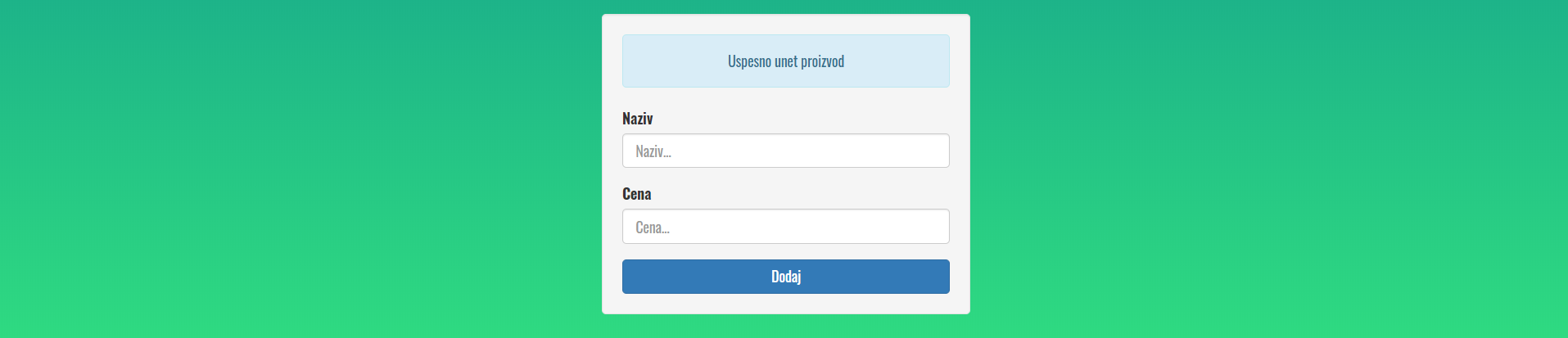
Adminu je posebno zanimljiv deo gde može da prati statistiku prodaje po danima.



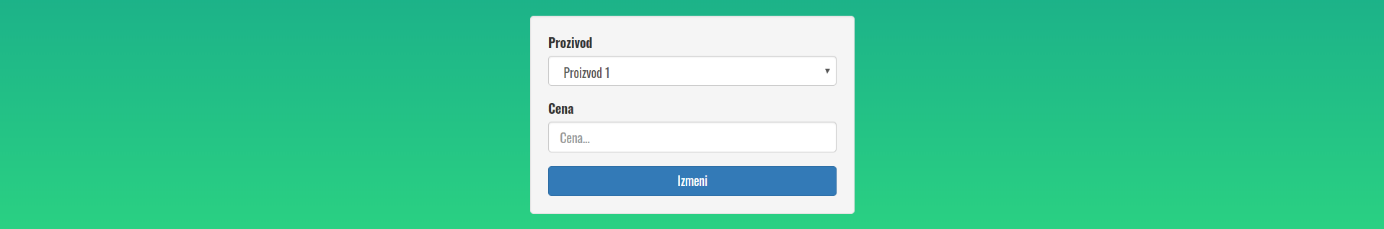
Klikom na dugme „Dodaj“ adminu se otvara prozor za unos novog proizvoda na sajt.



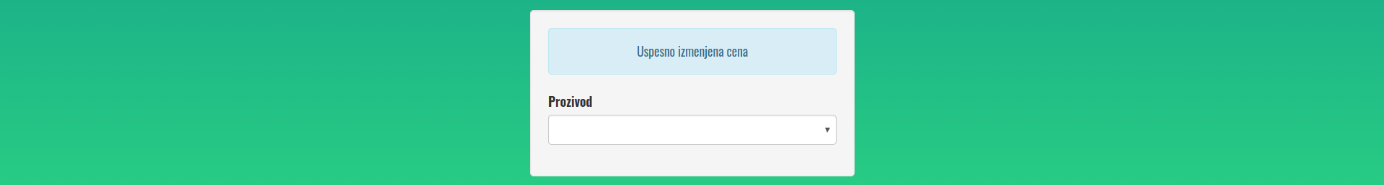
Kada admin uspešno doda proizvod pojavi mu se poruka.



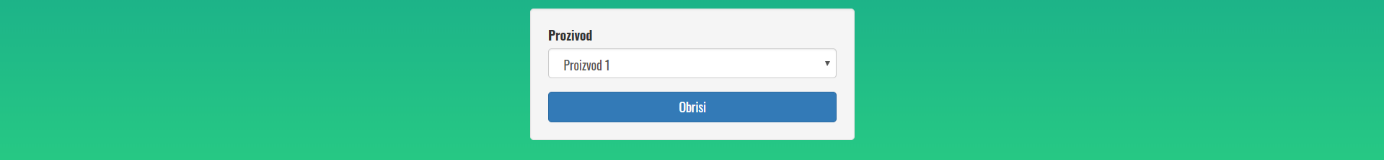
Klikom na dugme „Izmeni“ adminu se otvara prozor za izmenu proizvoda na sajtu.



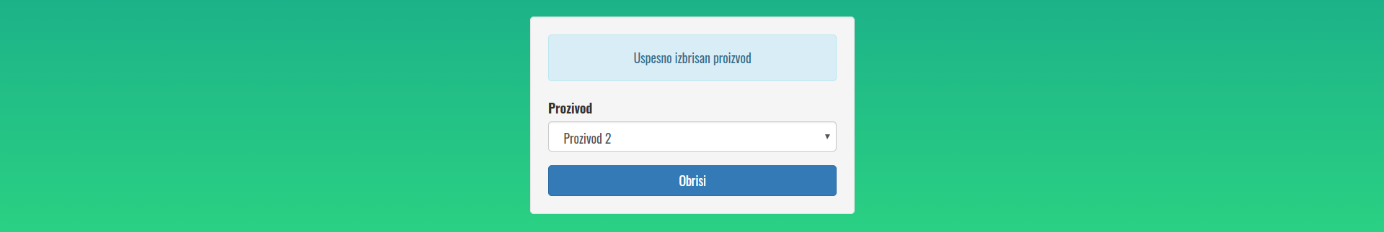
Iz padajućeg menija admin bira proizvod kome želi da promeni cenu i unosi novu cenu. Nakon unosa nove cene i kliktanja dugmeta Izmeni, admin dobija poruku u uspešnoj izmeni.



Admin ima mogućnost i da obriše proizvod.



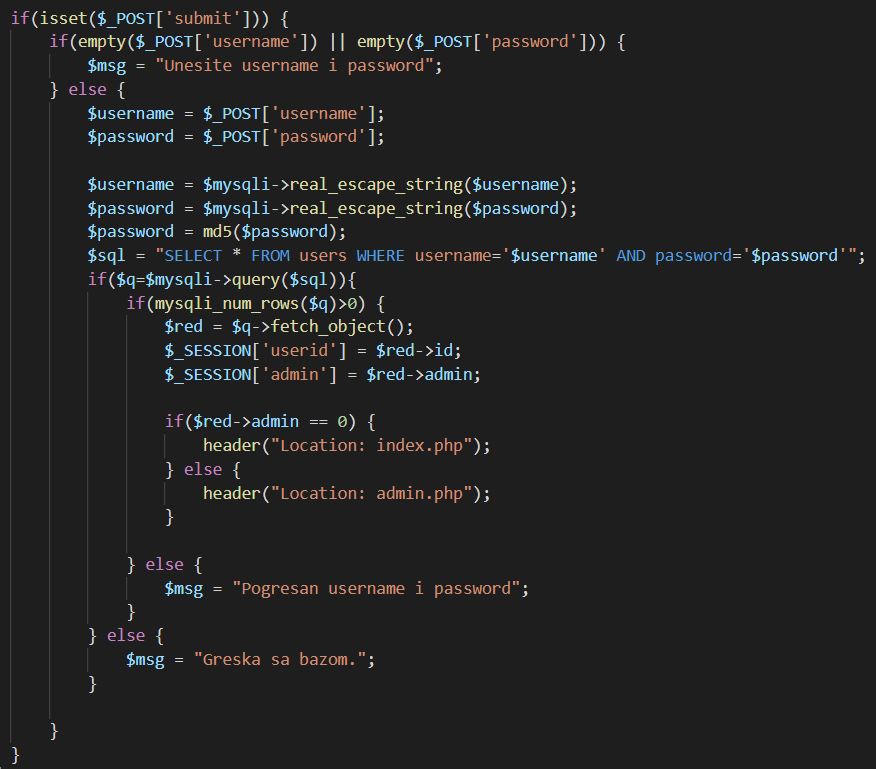
Admin iz padajućeg menija bira proizvod koji želi da obriše i kada ga obriše dobije poruku.



# Prikaz reprezentativnih delova koda

## 5.1 Login

Prijavljivanje korisnika



## 5.2 Logout

Sesija (index.php)

<?php

session\_start();

if(!isset($\_SESSION['userid'])) {

  header("Location: login.php");

}

Logout dugmence (logout.php) – zatvaramo sesiju i vraćamo se na stranicu login.php

<?php

    session\_start();

    session\_destroy();

    header("Location: login.php");

 ?>

Korisnik mora da bude ulogovan da bi pristupio stranici

<?php

session\_start();

if(!isset($\_SESSION['userid'])) {

  header("Location: login.php");

}

Zaštita aplikacije od SQL injection-a:

      $naziv = $mysqli->real\_escape\_string($naziv);

      $cena = $mysqli->real\_escape\_string($cena);

## 5.3 Primena datatables

(svi\_proizvodi.php)

CSS

  <link rel="stylesheet" href="https://cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css">

JavaScript datatables

  <script src="https://cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js"></script>

JQuery

  <script>

    $(document).ready( function () {

      $('table').DataTable({

         "language": {

                "url": "https://cdn.datatables.net/plug-ins/1.10.20/i18n/Serbian\_latin.json"

            }

      });

  } );

  </script>

## 5.4 Vizuelizacija podataka – Google Chart (Javni veb servis)

Google Chart (grafik.php)

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js" integrity="sha384-aJ21OjlMXNL5UyIl/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd" crossorigin="anonymous"></script>

  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

  <script>

    google.charts.load('current', {packages: ['corechart', 'bar']});

    setTimeout(function(){

      $.get( "controller.php?akcija=prodatBrojProizvodaPoDanima", function( json ) {

        var data = JSON.parse(json);

        var json = '{"cols":[{"label":"Datum","type":"string"},{"label":"Vrednost","type":"number"} , {"type":"number", "role":"interval"} , {"type":"number", "role":"interval"}],"rows":[';

        $.each( data, function( key, value ) {

          if(key==data.length-1) {

            json+= '{"c":[{"v":' + "\""+value.date+"\""+ '},{"v":'+value.num + '}]}';

          } else {

            json+= '{"c":[{"v":' + "\""+value.date+"\""+ '},{"v":'+value.num + '}]},';

          }

        });

         json+=']}';

              // Create our data table out of JSON data loaded from server.

        var data = new google.visualization.DataTable(json);

        var view = new google.visualization.DataView(data);

        var chart = new google.visualization.ColumnChart(

        document.getElementById('chart\_div'));

        chart.draw(view);

      });

     }, 1000);

      $.getJSON( "http://ip.jsontest.com", function( json ) {

       $("#ip\_adresa").html(json.ip);

     });

  </script>

## 5.5 Javni veb servis

Ispisivanje IP adrese korisnika (jQuery AJAX funkcija)- vraća podatke u JSON formatu (grafik.php)

       <h1>Prodati proizvodi po danima <small>IP: <span id="ip\_adresa"></span></small></h1>

      $.getJSON( "http://ip.jsontest.com", function( json ) {

      $("#ip\_adresa").html(json.ip);

     });

## 5.6 Listanje proizvoda (JQuery)

Vraćanje proizvoda (controller.php)

    session\_start();

    include "konekcija.php";

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="vratiProizvode") {

        $sql="SELECT \* FROM products";

        if($q=$mysqli->query($sql)) {

            $niz = array();

            while($red = $q->fetch\_object()) {

                $niz[] = $red;

            }

            echo json\_encode($niz);

        } else {

            $odgovor['poruka'] = "Greska sa bazom";

            echo json\_encode($odgovor);

        }

Dodaj proizvod (index.php)

    $.get( "controller.php?akcija=vratiProizvode", function( json ) {

      var data = JSON.parse(json);

      var dollar\_rate = "<?php echo $dollar\_rate; ?>";

      $.each( data, function( key, value ) {

        $("table tbody").append("<tr><td>"+value.product\_name+"</td><td>"+value.price+" EUR ("+value.price\*dollar\_rate+" USD)</td><td><button class='btn btn-primary btn-sm' onclick='otvoriModal("+value.id+")'>Dodaj proizvod</button></td></tr>");

      });

    });

## 5.7 Količina proizvoda u modalu

Omogućavamo unos količine

        <div class="modal-body">

          <p> Unesite kolicinu:</p>

          <div class="form-group">

            <label for="kolicina">Kolicina</label>

            <input type="text" class="form-control" id="kolicina" name="kolicina" placeholder="Kolicina...">

          </div>

          <input type="text" class="form-control" id="product\_id" name="product\_id" style="display: none;">

        </div>

        <div class="modal-footer">

          <button type="button" class="btn btn-default" data-dismiss="modal">Zatvori</button>

          <button type="button" class="btn btn-primary" onclick="dodajProizvodUKorpu();">Dodaj proizvod</button>

        </div>

      </div>

Funkcija otvoriModal()

    function otvoriModal(id) {

      $("#product\_id").val(id);

      $(".modal").modal("show");

    }

Dugme koje poziva modal

 $("table tbody").append("<tr><td>"+value.product\_name+"</td><td>"+value.price+" EUR ("+value.price\*dollar\_rate+" USD)</td><td><button class='btn btn-primary btn-sm' onclick='otvoriModal("+value.id+")'>Dodaj proizvod</button></td></tr>");

## 5.8 Dodavanje proizvoda u korpu

Dodaj proizvod u korpu (controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="dodajProizvodUKorpu") {

        $podaci\_json = file\_get\_contents("php://input");

        $podaci = json\_decode($podaci\_json);

        if($\_SESSION['korpa'][$podaci->product\_id] = $podaci->kolicina) {

            $odgovor['poruka'] = "Proizvod je uspesno ubacen.";

            echo json\_encode($odgovor);

        } else {

            $odgovor['poruka'] = "Doslo je do greske";

            echo json\_encode($odgovor);

        }

    }

dodajProizvodUKorpu na klik dugmeta iz modala (index.php)

        <div class="modal-footer">

          <button type="button" class="btn btn-default" data-dismiss="modal">Zatvori</button>

          <button type="button" class="btn btn-primary" onclick="dodajProizvodUKorpu();">Dodaj proizvod</button>

        </div>

    function dodajProizvodUKorpu() {

      var kolicina = $("#kolicina").val();

      var product\_id = $("#product\_id").val();

      if(kolicina.length == 0) {

        alert("Unesite kolicinu!");

      } else {

        var proizvod = new Proizvod(kolicina, product\_id);

        var json\_proizvod = JSON.stringify(proizvod);

        $.post( "controller.php?akcija=dodajProizvodUKorpu", json\_proizvod , function( json ) {

          var data = JSON.parse(json);

          alert(data.poruka);

           $(".modal").modal("hide");

           $("#kolicina").val("");

        });

      }

    }

    function Proizvod(kolicina, product\_id) {

      this.kolicina = kolicina;

      this.product\_id = product\_id;

    }

## 5.9 Korpa

(controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="vratiProizvodeIzKorpe") {

        if(isset($\_SESSION['korpa'])) {

            $niz = array();

            $sql="SELECT \* FROM products";

            if($q=$mysqli->query($sql)) {

                $proizvodi = array();

                while($red = $q->fetch\_object()) {

                    $proizvodi[$red->id] = $red;

                }

                foreach ($\_SESSION['korpa'] as $key => $value) {

                    $odgovor['proizvod\_id'] = $key;

                    $odgovor['proizvod'] = $proizvodi[$key]->product\_name;

                    $odgovor['kolicina'] = $value;

                    $odgovor['cena'] = $proizvodi[$key]->price;

                    $odgovor['ukupno'] = floatval($proizvodi[$key]->price)\*$value;

                    $niz[] = $odgovor;

                }

                echo json\_encode($niz);

            } else {

                $odgovor['status'] = 0;

                $odgovor['poruka'] = "Greska sa bazom";

                echo json\_encode($odgovor);

            }

        } else {

            $odgovor['status'] = 0;

            $odgovor['poruka'] = "Nema proizvoda u korpi.";

            echo json\_encode($odgovor);

        }

    }

(korpa.php)

    function ucitajKorpu() {

      $.get( "controller.php?akcija=vratiProizvodeIzKorpe", function( json ) {

        var data = JSON.parse(json);

        $("table tbody").empty();

        if(data.status==0) {

          alert(data.poruka);

        } else {

          $.each( data, function( key, value ) {

            $("table tbody").append("<tr><td>"+value.proizvod+"</td><td>"+value.cena+" EUR</td><<td>"+value.kolicina+"</td>td><td>"+value.ukupno+" EUR</td><td><button class='btn btn-primary btn-sm' onclick='obrisiIzKorpe("+value.proizvod\_id+")'>Obrisi proizvod</button></td></tr>");

          });

        }

      });

    }

## 5.10 Slanje porudžbine

Adresa (korpa.php)

 <div class="modal-body">

          <p> Unesite adresu:</p>

          <div class="form-group">

            <label for="adresa">Adresa</label>

            <input type="text" class="form-control" id="adresa" name="adresa" placeholder="Adresa...">

          </div>

        </div>

Pošalji porudžbinu dugme (korpa.php)

<button type="button" class="btn btn-primary" onclick="posaljiPorudzbinu();">Posalji</button>

Funkcija posaljiPorudzbinu() (korpa.php)

    function posaljiPorudzbinu() {

      var adresa = $("#adresa").val();

      if(adresa.length == 0) {

        alert("Unesite adresu!");

      } else {

        var json\_proizvod = '{"adresa":"'+adresa+'"}';

        console.log(json\_proizvod);

        $.post( "controller.php?akcija=posaljiPorudzbinu", json\_proizvod , function( json ) {

          var data = JSON.parse(json);

          alert(data.poruka);

           $(".modal").modal("hide");

           $("#adresa").val("");

           ucitajKorpu();

        });

      }

    }

(controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="posaljiPorudzbinu") {

        $podaci\_json = file\_get\_contents("php://input");

        $podaci = json\_decode($podaci\_json);

        $adresa = $mysqli->real\_escape\_string($podaci->adresa);

        $sql="INSERT INTO orders (user\_id, time\_ordered, address) VALUES ('".$\_SESSION['userid']."', NOW(), '".$adresa."')";

        if($q=$mysqli->query($sql)){

            $order\_id = $mysqli->insert\_id;

            foreach ($\_SESSION['korpa'] as $key => $value) {

                    $sql2="INSERT INTO orders\_list (order\_id, product\_id, quantity) VALUES('".$order\_id."', '".$key."', '".$value."')";

                    $q2=$mysqli->query($sql2);

                }

            unset($\_SESSION['korpa']);

            $odgovor['status'] = 1;

            $odgovor['poruka'] = "Porudzbina je poslata!";

            echo json\_encode($odgovor);

        } else {

            $odgovor['status'] = 0;

            $odgovor['poruka'] = "Greska sa bazom".mysqli\_error($mysqli);

            echo json\_encode($odgovor);

        }

    }

Pošalji porudžbinu (korpa.php)

        <button class="btn btn-primary" onclick="otvoriModal();">Posalji porudzbinu</button>

Otvaranje modala za pošalji porudžbinu (korpa.php)

    function otvoriModal() {

      $(".modal").modal("show");

    }

## 5.11 Brisanje proizvoda iz korpe

učitajKorpu() (korpa.php)

    function ucitajKorpu() {

      $.get( "controller.php?akcija=vratiProizvodeIzKorpe", function( json ) {

        var data = JSON.parse(json);

        $("table tbody").empty();

        if(data.status==0) {

          alert(data.poruka);

        } else {

          $.each( data, function( key, value ) {

            $("table tbody").append("<tr><td>"+value.proizvod+"</td><td>"+value.cena+" EUR</td><<td>"+value.kolicina+"</td>td><td>"+value.ukupno+" EUR</td><td><button class='btn btn-primary btn-sm' onclick='obrisiIzKorpe("+value.proizvod\_id+")'>Obrisi proizvod</button></td></tr>");

          });

        }

      });

    }

Brisanje iz korpe (korpa.php)

    function obrisiIzKorpe(id) {

      $.get( "controller.php?akcija=obrisiProizvodIzKorpe&proizvod\_id="+id, function( json ) {

        var data = JSON.parse(json);

        alert(data.poruka);

        ucitajKorpu();

      });

    }

Obriši proizvode iz korpe (controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="obrisiProizvodIzKorpe") {

        $id\_proizvoda = $\_GET['proizvod\_id'];

        unset($\_SESSION['korpa'][$id\_proizvoda]);

        $odgovor['poruka'] = "Proizvod je uspesno obrisan iz korpe.";

        echo json\_encode($odgovor);

    }

## 5.12 Obrada porudžbina

Da bi se obradila porudžbina mora da bude ulogovan admin (admin.php)

<?php

session\_start();

if(!isset($\_SESSION['userid']) || $\_SESSION['admin'] !=1) {

  header("Location: login.php");

}

?>

vratiNeobradjenePorudzbine (controller.php)

   if(isset($\_GET['akcija']) && $\_GET['akcija']=="vratiNeobradjenePorudzbine") {

        $sql="SELECT o.\*, u.username FROM orders as o JOIN users as u ON o.user\_id = u.id WHERE status=0";

        if($q=$mysqli->query($sql)) {

            $niz = array();

            while($red = $q->fetch\_object()) {

                $niz[] = $red;

            }

            echo json\_encode($niz);

        } else {

            $odgovor['poruka'] = "Greska sa bazom";

            echo json\_encode($odgovor);

        }

    }

Funkcija UcitajNeobradjenePorudzbine() (admin.php)

    function ucitajNeobradjenePorudzbine() {

      $.get( "controller.php?akcija=vratiNeobradjenePorudzbine", function( json ) {

        var data = JSON.parse(json);

        $("#neobradjene\_porudzbine tbody").empty();

         $.each( data, function( key, value ) {

            $("#neobradjene\_porudzbine tbody").append("<tr><td>"+value.username+"</td><td>"+value.time\_ordered+" EUR</td><<td>"+value.address+"</td><td><button class='btn btn-primary btn-sm' onclick='prihvatiPorudzbinu("+value.id+")'>Prihvati</button></td></tr>");

          });

      });

    }

Ovo funkcija se poziva onload (admin.php)

<body onload="ucitajNeobradjenePorudzbine(); ucitajObradjenePorudzbine();">

Dugme Prihvati poziva funkciju prihvatiPorudzbinu() (admin.php)

    function prihvatiPorudzbinu(id) {

      $.get( "controller.php?akcija=promeniStatusPorudzbine&order\_id="+id, function( json ) {

        ucitajObradjenePorudzbine();

        ucitajNeobradjenePorudzbine();

      });

    }

vratiObradjenePorudzbine (controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="vratiObradjenePorudzbine"){

        $sql="SELECT o.\*, u.username FROM orders as o JOIN users as u ON o.user\_id = u.id WHERE status=1";

        if($q=$mysqli->query($sql)) {

            $niz = array();

            while($red = $q->fetch\_object()) {

                $niz[] = $red;

            }

            echo json\_encode($niz);

        } else {

            $odgovor['poruka'] = "Greska sa bazom";

            echo json\_encode($odgovor);

        }

    }

Funkcija UcitajObradjenePorudzbine() (admin.php)

    function ucitajObradjenePorudzbine() {

      $.get( "controller.php?akcija=vratiObradjenePorudzbine", function( json ) {

        var data = JSON.parse(json);

        $("#obradjene\_porudzbine tbody").empty();

         $.each( data, function( key, value ) {

            $("#obradjene\_porudzbine tbody").append("<tr><td>"+value.username+"</td><td>"+value.time\_ordered+" EUR</td><<td>"+value.address+"</td><td>Prhivaceno</td></tr>");

          });

      });

    }

Ovo funkcija se poziva onload (admin.php)

<body onload="ucitajNeobradjenePorudzbine(); ucitajObradjenePorudzbine();">

Na dugme Prihvati menjamo status narudžbini tačnije od neobrađene narudžbina postaje obrađena. Koristimo promeniStatusPorudzbine (controller.php)

    if(isset($\_GET['akcija']) && $\_GET['akcija']=="promeniStatusPorudzbine") {

        $order\_id = $\_GET['order\_id'];

        $sql="UPDATE orders SET status=1 WHERE id='$order\_id'";

        if($q=$mysqli->query($sql)) {

            $odgovor['poruka'] = "Promenjen status porudzbine.";

            echo json\_encode($odgovor);

        } else {

            $odgovor['poruka'] = "Greska sa bazom";

            echo json\_encode($odgovor);

        }

    }

Funkcija prihvatiPorudzbinu() (admin.php)

    function prihvatiPorudzbinu(id) {

      $.get( "controller.php?akcija=promeniStatusPorudzbine&order\_id="+id, function( json ) {

        ucitajObradjenePorudzbine();

        ucitajNeobradjenePorudzbine();

      });

    }

# GitHib link ka repozitorijumu

https://github.com/1057-18/itehProjekat1.git