Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

E- Cvećara

Seminarski rad iz predmeta

Internet tehnologije

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Filip | Đalić | 2019/321 |
| Katarina | Grujić | 2019/195 |
| Link ka Github-u | <https://github.com/KacaGrujic/projekatcvecara.git> | |

Sadržaj

[Uvod 3](#_Toc140102790)

[1 Korisnički zahtev 4](#_Toc140102791)

[2. Opis sistema 5](#_Toc140102792)

[2.1. Slučajevi korišćenja 5](#_Toc140102793)

[2.2. Opis arhitekture aplikacije 7](#_Toc140102794)

[2.3. Opis procesa slučajeva korišćenja (dijagrami sekvenci) 11](#_Toc140102795)

[2.4 Model podataka-Dijagram klasa, struktura baze 13](#_Toc140102796)

[2.5. Specifikacija REST API-ja 15](#_Toc140102797)

[3. Opis tehnologija korišćenih u aplikaciji 22](#_Toc140102798)

[JavaScript 22](#_Toc140102799)

[JSON 23](#_Toc140102800)

[React 23](#_Toc140102801)

[Laravel 24](#_Toc140102802)

[MVC Pattern 25](#_Toc140102803)

[4. Korisničko uputstvo 25](#_Toc140102804)

[5. Prikaz reprezentativnih delova koda 34](#_Toc140102805)

[6. Github link ka repozitorijumu 52](#_Toc140102806)

# Uvod

Full stack web aplikacija “Blossom Blooms” predstavlja e-cvećaru implementiranu pomoću Laravel i React tehnologija.

U prvom poglavlju, opisani su korisnički zahtevi i navedeni slučajevi korišćenja. Takođe, navedene su funkcionalnosti aplikacije.

U drugom poglavlju, izdvojeni su karakteristični slučajevi korišćenja, kao i dijagrami slučajeva korišćenja za aktore Admin i Korisnik.

Takođe, izdvojene su glavne komponente aplikacije, objašnjen je način komunikacije komponenti međusobno i sa bazom

Za karakteristične slučajeve korišćenja navedeni su dijagrami sekvenci kao opis procesa. Model podataka predstavljen je pomoću dijagrama klasa, a predstavljena je i struktura baze podataka.

Tabelarno su prikazane funkcije sopstvenog web servisa, njihovi ulazni i izlazni podaci, format podataka, url rute kao i ulazni i izlazni parametri.

U trećem poglavlju, ukratko su opisane tehnologije korišćene u implementaciji.

U četvrtom poglavlju, prikazani su karakteristični slučajevi korišćenja preko korisničkog interfejsa, screenshot-ovi aplikacije i korak-po-korak prikaz slučajeva korišćenja.

U petom poglavlju, izdvojeni su karakteristični delovi koda.

U šestom poglavlju nalazi se link ka github repozitorijumu.

# Korisnički zahtev

Potrebno je projektovati i implementirati veb aplikaciju virtuelne cvećare „Blossom Blooms“. Aplikacija je osmišljena tako da korisnik može vrlo lako naručiti buket tako što će se ulogovati na sajt, odabrati željene proizvode i naručiti. Sistem ima dva nivoa pristupa: admin i korisnici(kupci). Svi korisnici moraju biti registrovani i prijavljeni na sistem pod email-om i lozinkom (funkcionalnosti Registracija, Login, Logout). Admin ima pristup funkcionalnostima prikaza podataka iz baze (tabele flowers, messages, order) i ima nadležnost manipulacije podacima (funkcionalnosti: Dodavanje, Brisanje, Izmena cveća). On ujedno može videti i pristigle poruke koje korisnici šalju putem kontakt forme. Pored ovoga, admin ima uvid u statističke podatke, odosno cene proizvoda i koliko ih je u ponudi. Korisnici koji imaju ulogu kupca na našem sajtu imaju pristup sledećim funkcionalnostima: Dodavanje proizvoda u korpu i brisanje proizvoda iz iste, mogu odabrati valutu u kojoj žele da bude izražena cena porudžbine i samim tim i da generišu račun. Kao što je prethodno navedeno, korisnik može poslati poruku putem kontakt forme.

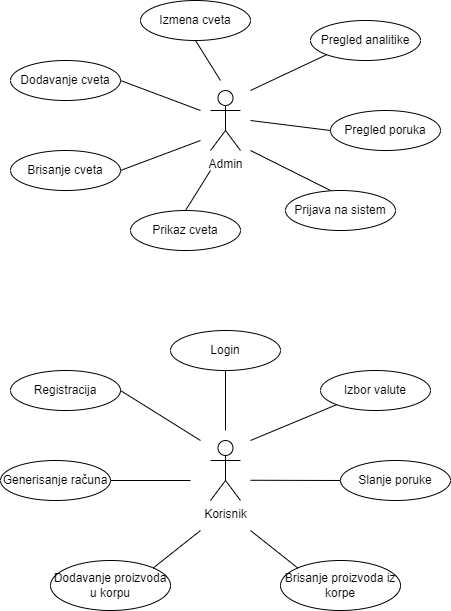
Slučajevi korišćenja:

1. Registracija
2. Login
3. Logout
4. Prikaz cveta
5. Izmena cveta
6. Brisanje cveta
7. Dodavanje proizvoda u korpu
8. Brisanje proizvoda iz korpe
9. Izbor valute
10. Generisanje računa
11. Slanje poruke
12. Pregled poruka
13. Pregled analitike

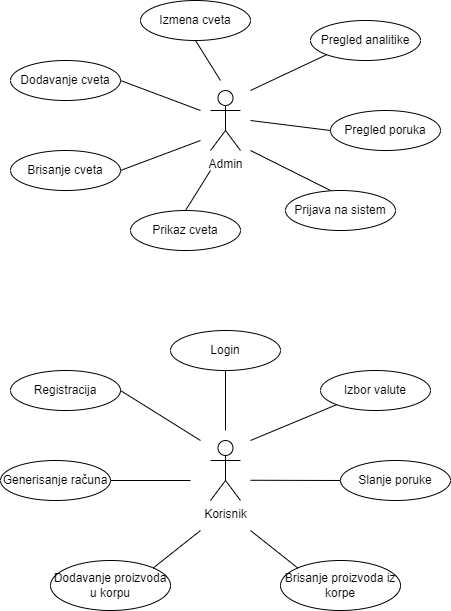
# Opis sistema

## Slučajevi korišćenja

Dijagram slučajeva korišćenja za Admina:



Dijagram slučajeva korišćenja za Korisnika:



1.Slučaj korišćenja: Dodavanje cveta(buketa)

Akter: Admin

Preduslov:Sistem je pokrenut i Admin je ulogovan sa email-om i lozinkom.

Postuslov: Kreiran je cvet.

Osnovni scenario:

1. Admin pokreće formu za unos podataka za cvet klikom na dugme Add flower na stranici Admin.
2. Sistem instancira potrebne objekte i prikazuje formu dodavanja cveta.
3. Admin unosi podatke o cvetu: ime, opis, cenu, sliku.
4. Admin bira iz padajućeg menija vrednost za kategoriju.
5. Admin pritiska dugme „Add“ i poziva sistem da obradi zahtev.
6. Sistem prihvata zahtev i obezbeđuje čuvanje objekta u bazu pod transakcijom.

Alternativni scenario:

5.1. Sistem saopštava adminu da unos nije validan. (IA)

2.Slučaj korišćenja: Dodavanje proizvoda u korpu

Akter: Korisnik

Preduslov: Korisnik je ulogovan na sistem sa email-om i lozinkom.

Postuslov: U korpi su prikazane naručeni proizvodi.

Osnovni scenario:

1. Udomitelj pokreće stranicu „Shop“.
2. Udomitelj bira buket iz niza.
3. Udomitelj klikom na dugme „+“ poziva sistem da obradi zahtev.
4. Sistem prihvata zahtev i pamti iznos porudžbine.
5. Sistem prikazuje odabrani buket i cenu na stranici „Cart“.

3.Slučaj korišćenja: Slanje poruke

Akter: Korisnik

Preduslov: Korisnik je ulogovan na sistem sa email-om i lozinkom.

Postuslov: Poruka je poslata.

Osnovni scenario:

1. Korisnik pokreće formu za unos poruke klikom na dugme Contact na navigacionom meniju.
2. Sistem instancira potrebne objekte i prikazuje formu za kontakt.
3. Admin unosi podatke:email, ime i poruku.
4. Admin pritiska dugme „Send“ i poziva sistem da obradi zahtev.
5. Sistem prihvata zahtev i obezbeđuje čuvanje objekta u bazu pod transakcijom.

## 2.2. Opis arhitekture aplikacije

Aplikacija se sastoji iz dve komponente: laravel projekat i React projekat. Laravel projekat je back-end aplikacije,a React projekat front-end.Arhitektura je tronivojska i zasnovana na MVC paternu; sastoji se korisničkog interfejsa, aplikacione logike i lokalne baze podataka.

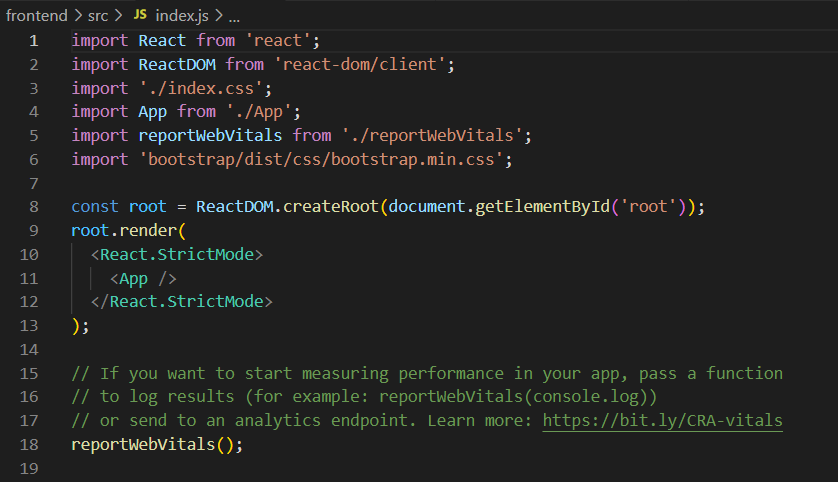
Da bi se povezala aplikacija sa bazom podataka, back-end komponenta obično koristi drajver baze podataka ili biblioteku za mapiranje objekata na relacijsku bazu podataka (Object-Relational Mapping - ORM). Drajver baze podataka ili ORM biblioteka omogućavaju back-end-u da šalje SQL upite ili da izvršava operacije nad bazom podataka programski iz back-end koda.

U suštini, povezivanje između full-stack aplikacije i baze podataka uključuje konfigurisanje drajvera baze podataka ili ORM biblioteke na back-end-u, uspostavljanje konekcije sa bazom podataka koristeći detalje konekcije, i izvršavanje operacija nad bazom podataka iz serverskog koda kao odgovor na interakcije korisnika.

Korisnički interfejs predstavlja ulazno-izlaznu reprezentaciju veb aplikacije. Aplikaciona logika nema znanja o tome gde se nalazi korisnički interfejs. Aplikaciona logika sadrži deo za komunikaciju sa klijentima, deo za komunikaciju sa skladištem i sadrži poslovnu logiku. Ona je implementirana u Laravel projektu i čine je kontroleri apl. Logike: *FlowerController, OrderController, OrderItemController, UserController.* Skladište podataka (baza podataka) nalazi se na localhost/phpMyAdmin i čuva stanja atributa.

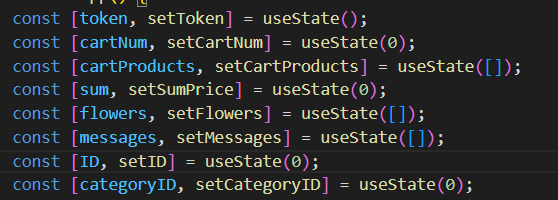
React projekat- Osnovni segmenti:

Root komponenta u React projektu je App.js, ona predstavlja parent komponentu za sve ostale i renderuje se. Renderovanjem App.js komponente dobijamo kompletan sajt, sa svim stranicama. App.js se poziva u render metodi u fajlu index.js:



**App.js** komponenta sadrži sve ostale glavne child komponente, a to su:Login.js, Register.js, AdminView, Home.js, NavBar.js, Footer.js, Flower.js, Contact.js, Cart.js i ostale. Svojstvo koje se koristilo za učitavanje različitih komponenti jeste **rutiranje** i na taj način su komponente međusobno povezane**,** pri čemu se prvo proverava da li je korisnik admin ili ne i na osnovu toga sledi prikaz komponente(ili stranice).

Komponente međusobno komuniciraju i prosleđenim parametrima koji se tiču podataka( npr. flowers, cartProducts, cartNum, sum,..) Vrednosti podataka se postavljaju korišćenjem useState hooks:



*windows.session.storage*predstavlja način da web developer-i čuvaju podatke tokom korisnikove sesije, čime dobijamo dinamičnu i interaktivnu web aplikaciju. U ovom slučaju, u windows.session.storage u Inspect-Network prikazana je vrednost tokena prijavljenog korisnika i vrednost isAdmin polja što našoj aplikaciji govori da li je korisnik prijavljen i da li je njegovo svojstvo admin ili ne.

Laravel projekat:

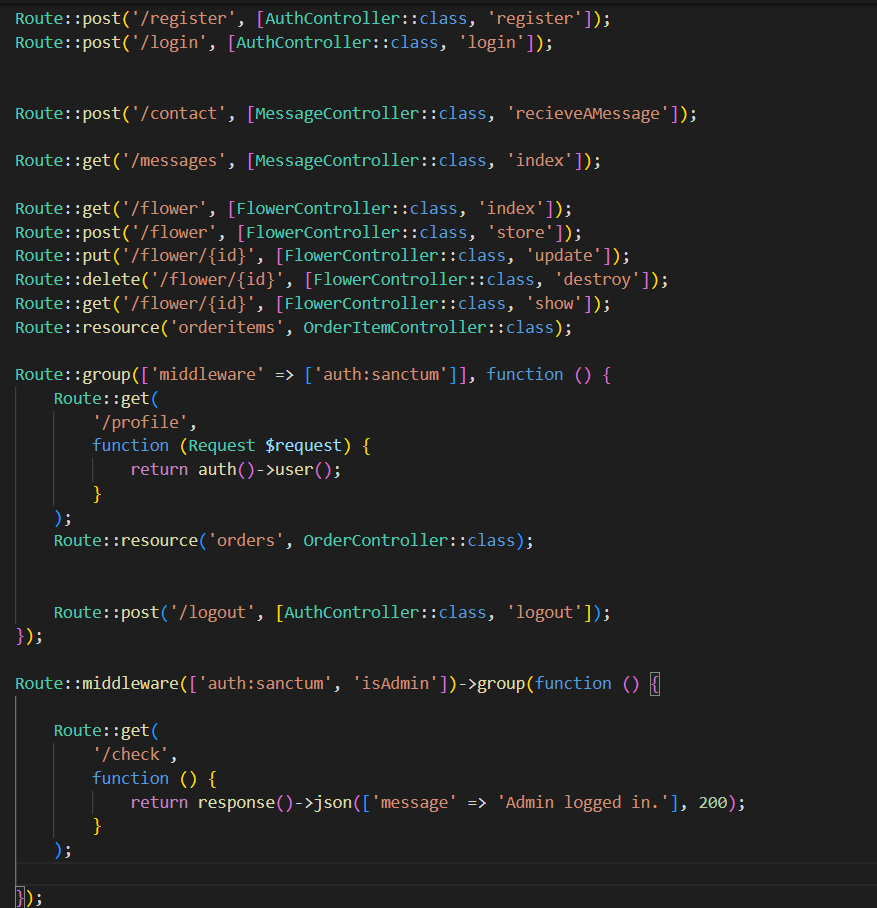
Glavne komponente su modeli, njihovi kontroleri, migracije I rute. Migracije su koncept u Laravel-u pomoću kog se kreiraju, brišu ili menjaju tabele u bazi podataka, bez korišćenja sql upita već samo kroz PHP kod. Migracije su u PHP definisane kao klase (database/migrations). Migracije su direktna komunikacija sa DB brokerom**.** Modeli su Flower.php, User.php, Order.php, OrderItem.php, Cateogory.php, Message.php i svaki model se povezuje sa tačno jednom tabelom u bazi podataka. Modeli sadrže svojstva tabele(atribute tj. kolone tabele), kao i ograničenja nad manipulacijom podacima (npr. Fillable svojstvo). Takođe, u modelima se nalaze funkcije koje oslikavaju vezu između modela svojstvima belongsTo, hasOne, hasMany.



Svaki model ima svog kontrolora koji je zadužen za komunikaciju sa bazom. Svaki kontroler ima funkcije koji predstavljaju aplikacionu logiku.

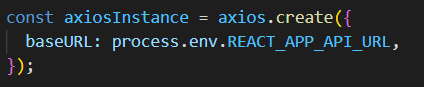
Najznačajni koncept Laravel-a jesu rute. Svaka ruta definiše URL, HTTP metodu i akciju koju treba izvršiti kada se pristupi tom URL-u. Akcija može biti anonimna funkcija, poziv na metodu kontrolera ili zatvaranje ("closure"). Laravel takođe podržava definisanje ruta s parametrima, grupa ruta, definisanje ruta s nazivima, ...Rute su ključni koncept u Laravel-u koji omogućava programerima da definišu kako aplikacija obrađuje ulazne zahteve.

Primeri različitih ruta:



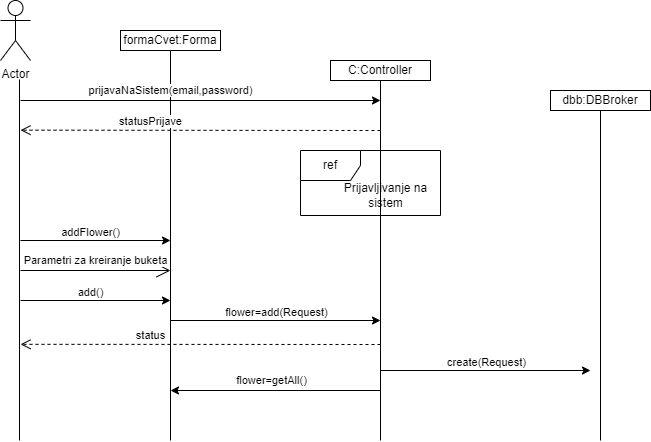
Rute u Laravel-u predstavljaju način za povezivanje sa React-om. Laravel takođe pruža opcije za prenos podataka iz PHP u JavaScript, što je korisno za deljenje podataka između serverskog i klijentskog dela aplikacije, pri čemu Laravel projekat predstavlja serverski, a React projekat klijentski deo aplikacije.

Axios je JavaScript biblioteka za upravljanje HTTP zahtevima od JavaScript klijenta ka serveru, ali i upravlja odgovorima od serverskoj ka klijentskoj strani.

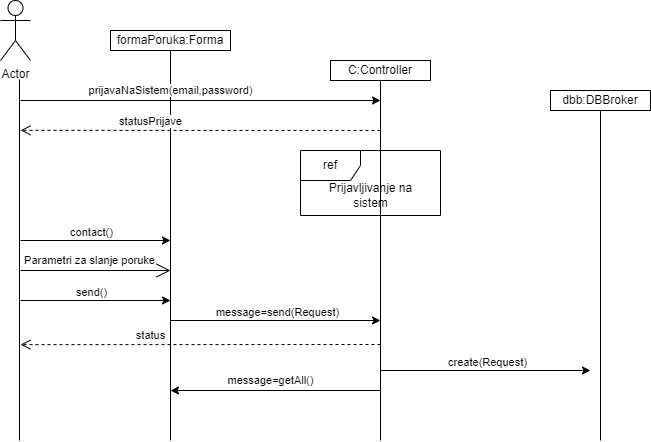


## 2.3. Opis procesa slučajeva korišćenja (dijagrami sekvenci)

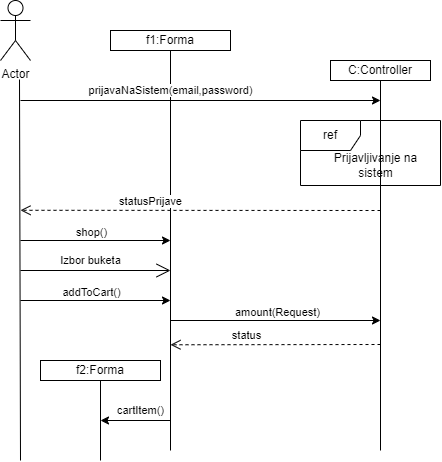
Dijagram sekvenci za slučaj korišćenja Dodavanje cveta(buketa):



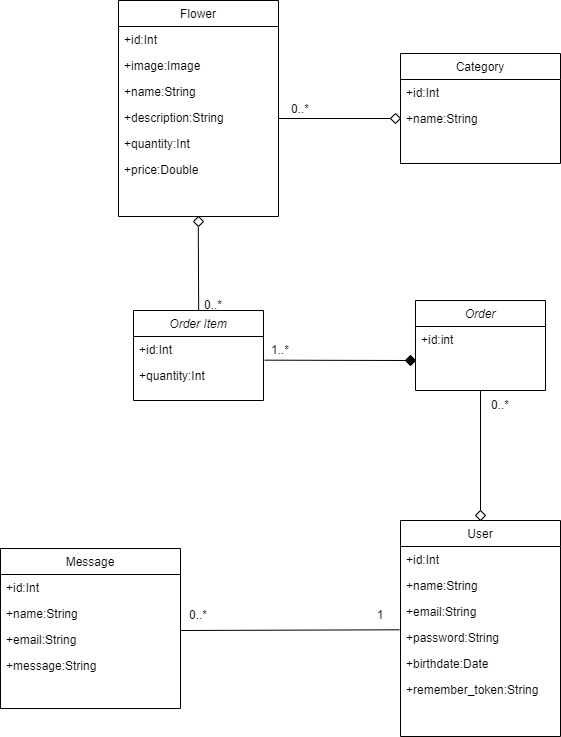
Dijagram sekvenci za slučaj korišćenja Slanje poruke:



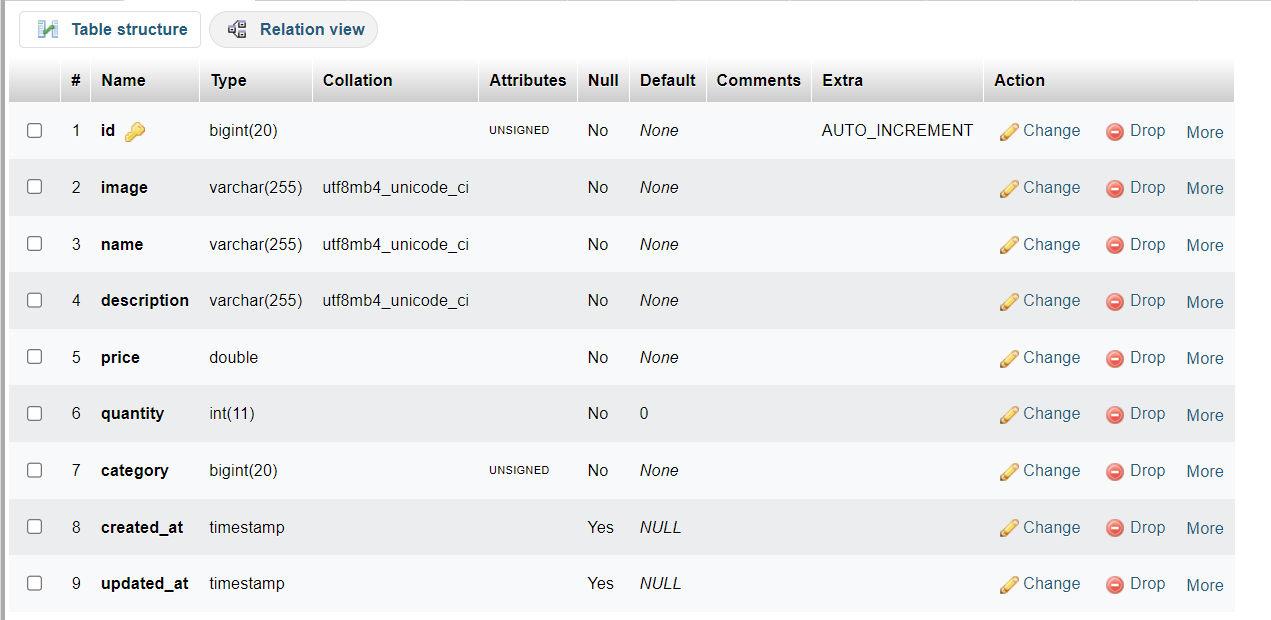
Dijagram slučajeva korišćenja za Dodavanje proizvoda u korpu:

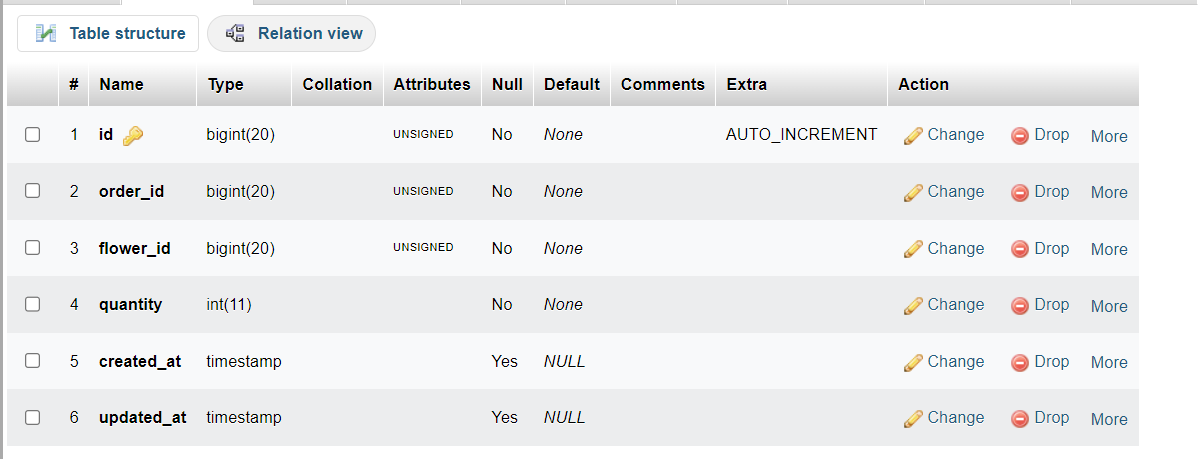


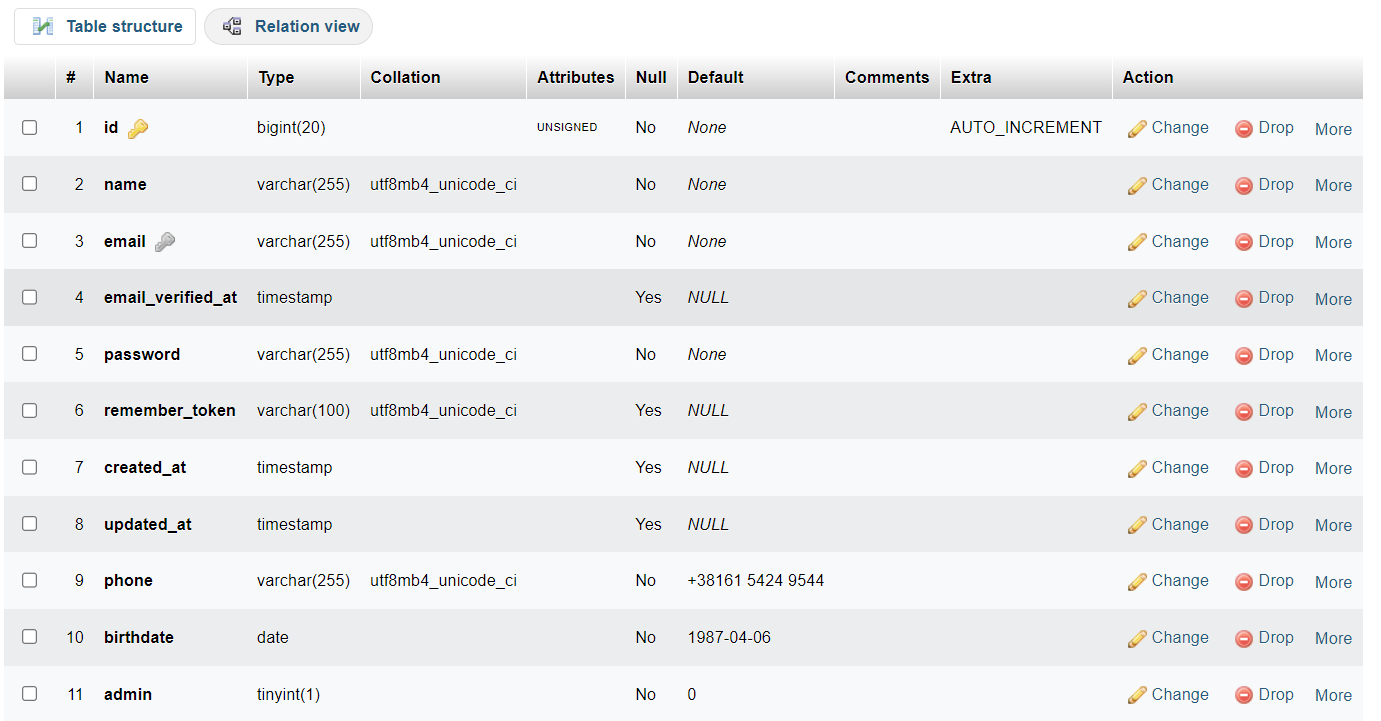
## 2.4 Model podataka-Dijagram klasa, struktura baze

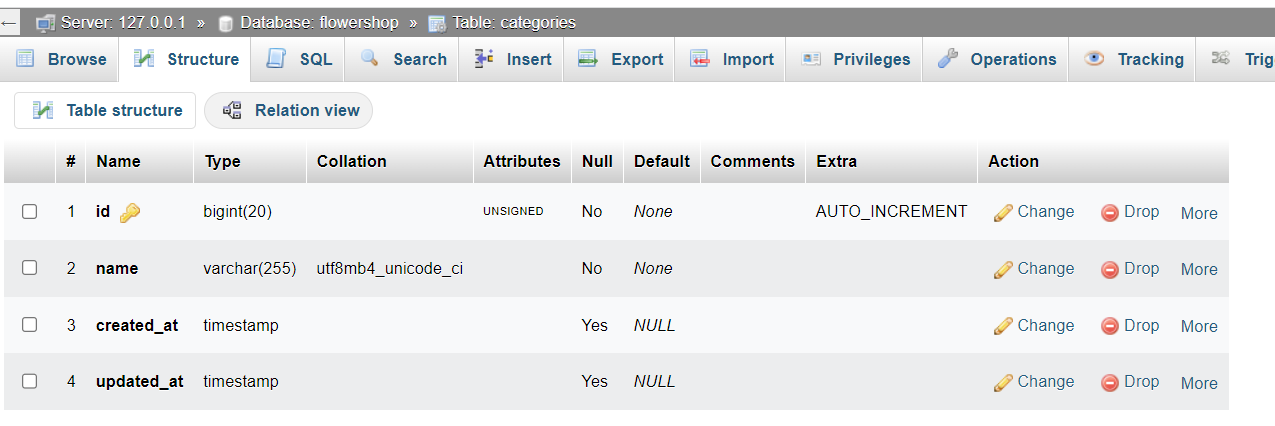


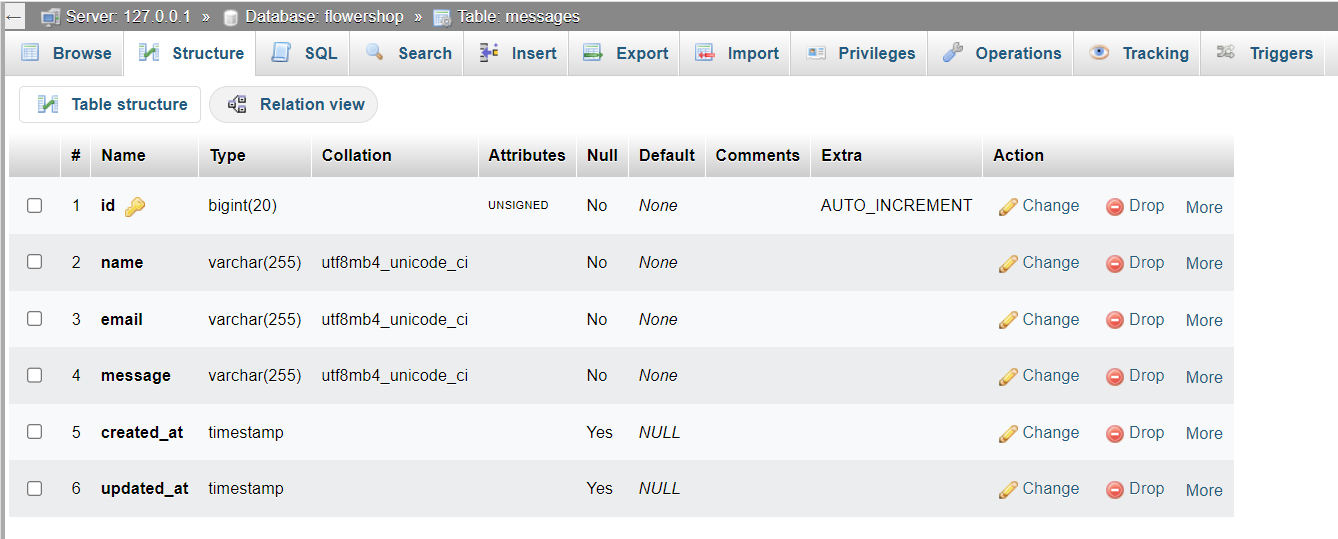
Struktura baze:











## 2.5. Specifikacija REST API-ja

|  |  |
| --- | --- |
| Opis funkcije | Registrovanje korisnika u sistema |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | Nema |
| HTTP body parametri | fullname, email, password,phone,birthdate |
| Format HTTP body parametara | {      "name" : "Filip",      "email": "filip123@gmail.com",      "password" : "filp123"      "phone" : "0644122563"      "birthdate" : "30/07/2000."  } |
| Izlazni parametri | {      "data": {          "name": "Filip",          "email": "filip123@gmail.com",          "updated\_at": "2023-06-27T13:04:26.000000Z",          "created\_at": "2023-06-27T13:04:26.000000Z",          "id": 12      },      "acess\_token": "1|evOQQDA6zUbmZgns6oYse993gbeQNBv9FaSJdFh8",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Logovanje korisnika na sistem |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | Nema |
| HTTP body parametri | email, password |
| Format HTTP body parametara | {      "email" : " filip123@gmail.com ",      "password" : "filip123"  } |
| Izlazniparametri | {      {      "0": {          "status": 200,          "username": "Filip",          "token": "33|7zPsrq6wF0Gg5EHjvGTqQsfiCnceAxLUtXrY0tZC",          "role": " ",          "id": 12      },      "success": **true**  }  } |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Odjavljivanje korisnika sa Sistema |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazni parametri | {   "You have successfully logged out!"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz poruka |
| HTTP metoda | GET |
| URL | /api/poruke |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazni parametri | {          "id": 2,          "name": "Nikola",          "email": "nikola@gmail.com",          "message": "Da li isporucujete bukete i van drzave?",          "created\_at": "2023-07-10T00:48:35.000000Z",          "updated\_at": "2023-07-10T00:48:35.000000Z"      }, |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Prikaz cveća(buketa) |
| HTTP metoda | GET |
| URL | /api/flower |
| URL parametri | nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | {      "data": [          {              "id": 1,              "name": "Buket ruza",              "description": "Buket je napravljen sa pet crvenih holandskih ruža i dekoracijom. Ukoliko želite da nekoga iznenadite i pošaljete na adresu mali znak pažnje, ovaj buket je idealan poklon.",              "price": 3000,              "quantity": 0,              "category": {                  "id": 2,                  "name": "ruze"              },              "image": "https://www.cvecarakazablanka.com/data/products/1736/feature/0/Buket\_-\_pet\_crvenih\_ru%C5%BEa\_sa\_dekoracijom.jpg"          },          {              "id": 2,              "name": "Buket irisa",              "description": "Buket je napravljen sa žutim gerberima i plavim irisima koji su poput leptirića koji slete na mirisno lepo cveće. Iznenadite dragu osobu simpatičnim buketom i ulepšajte joj dan.",              "price": 1950,              "quantity": 0,              "category": {                  "id": 6,                  "name": "irisi"              },              "image": "https://www.cvecarakazablanka.com/data/products/1325/feature/0/buket\_27.jpg"          },          {              "id": 3,              "name": "Buket gerbera",              "description": "Buket gerbera namenjen je samo za prave dame, one sa ukusom. Buket je pretežno sačinjen od roze gerbera, upotpunjuju ga i ruže, lizijantus, orjentalni ljiljan kao i sezonsko zelenilo. Dajte doznanja da vaša dama nikada nije sama.",              "price": 5200,              "quantity": 0,              "category": {                  "id": 4,                  "name": "gerberi"              },              "image": "https://www.cvecaralotos.rs/wp-content/uploads/2020/12/cvecara-pozarevac.jpg"          },          {              "id": 4,              "name": "Buket suncokreta",              "description": "Buket je napravljen sa suncokretima koji su simbol sunčeve svetlosti i energije. Ukoliko želite da dragoj osobi ulepšate dan buketom veselih boja, pošaljite joj isti na kućnu adresu i izmamite joj osmeh na lice. ",              "price": 3850,              "quantity": 0,              "category": {                  "id": 1,                  "name": "suncokreti"              },              "image": "https://www.cvecarakazablanka.com/data/products/1323/feature/0/24\_buket.jpg"          }      ]  } |
| Format izlaznihparametara | application/json |
|  |  |
| Opis funkcije | Dodavanje novog cveta(buketa) |
| HTTP metoda | POST |
| URL | /api/flower |
| URL parametri | Nema |
| HTTP body parametri | Name, description, price, quantity, category, image |
| Format HTTP body parametara | {          "name": "Buket lala 3",          "description": "afgadgdg",          "price": 8100,          "quantity": 0,          "category": 2,          "image": "https://cvecar.com/wp-content/uploads/buket-25-ruza.jpg"  } |
| Izlazni parametri | [      "Flower succcessfully added",      {          "name": "Buket lala 3",          "price": 8100,          "description": "afgadgdg",          "quantity": 0,          "category": 2,          "image": "https://cvecar.com/wp-content/uploads/buket-25-ruza.jpg",          "updated\_at": "2023-07-11T21:11:40.000000Z",          "created\_at": "2023-07-11T21:11:40.000000Z",          "id": 10      }  ] |
| Format izlaznihparametara | application/json |
|  |  |
| Opis funkcije | Prikazivanje jednog proizvoda |
| HTTP metoda | GET |
| URL | /api/flower/{flowerId} |
| URL parametri | flowerId |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazni parametri | {      "data": {          "id": 2,          "name": "Buket irisa",          "description": "Buket je napravljen sa žutim gerberima i plavim irisima koji su poput leptirića koji slete na mirisno lepo cveće. Iznenadite dragu osobu simpatičnim buketom i ulepšajte joj dan.",          "price": 1950,          "quantity": 0,          "category": {              "id": 6,              "name": "irisi"          },          "image": "https://www.cvecarakazablanka.com/data/products/1325/feature/0/buket\_27.jpg"      }  } |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Prikazivanje stavki porudžbine |
| HTTP metoda | GET |
| URL | /api/orderitems |
| URL parametri | nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | [      {          "id": 2,          "order\_id": 12,          "flower\_id": 2,          "quantity": 3,          "created\_at": "2023-07-10T21:44:24.000000Z",          "updated\_at": "2023-07-10T21:44:24.000000Z"      },      {          "id": 3,          "order\_id": 12,          "flower\_id": 6,          "quantity": 3,          "created\_at": "2023-07-10T21:44:24.000000Z",          "updated\_at": "2023-07-10T21:44:24.000000Z"      }  ] |
| Format izlaznihparametara | application/json |
|  |  |
| Opisfunkcije | Brisanje odabranog cveta(buketa) |
| HTTP metoda | DELETE |
| URL | /api/flower/{flowerId} |
| URL parametri | flowerId |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | "Flower is successfully deleted!" |
| Format izlaznih parametara | application/json |
|  |  |
| Opisfunkcije | Ažuriranje cveta(buketa) |
| HTTP metoda | PUT/PATCH |
| URL | /api/flower/{flowerId} |
| URL parametri | flowerId |
| HTTP body parametri | Name, description, price, quantity, category, imge |
| Format HTTP body parametara | {          "name": "Buket lala 3",          "description": "Ovo je novi opis proizvoda",          "price": 8100,          "quantity": 0,          "category": 2,          "image": "https://cvecar.com/wp-content/uploads/buket-25-ruza.jpg"    } |
| Izlazni parametri | [      "Flower is successfully updated!",      {          "id": 9,          "name": "Buket lala 3",          "description": "Ovo je novi opis proizvoda",          "price": 8100,          "quantity": 0,          "category": {              "id": 2,              "name": "ruze"          },          "image": "https://cvecar.com/wp-content/uploads/buket-25-ruza.jpg"      }  ] |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Slanje poruka |
| HTTP metoda | POST |
| URL | /api/contact |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | {              "name": "Filip",              "message": "Koje kartice prihvatate?",              "email": "filip123@gmail.com"   } |
| Izlazni parametri | {      "success": **true**  } |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Prikazivanje profila |
| HTTP metoda | GET |
| URL | /api/profile |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazni parametri | {      "id": 12,      "name": "Filip",      "email": "filip123@gmail.com",      "email\_verified\_at": **null**,      "created\_at": "2023-06-27T13:04:26.000000Z",      "updated\_at": "2023-02-27T13:04:26.000000Z",      "phone" : "0644122563",      "birthdate" : "30/07/2000.",      "admin": 0  } |
| Format izlaznih parametara | application/json |
| Opisfunkcije | Admin ulogovan |
| HTTP metoda | GET |
| URL | /api/check |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazni parametri | {      "message": "Admin logged in"  } |
| Format izlaznih parametara | application/json |

# Opis tehnologija korišćenih u aplikaciji

U ovoj aplikaciji korišćene su sledeće tehnologije:

* HTML
* CSS
* JavaScript
* Laravel
* React
* JSON

## JavaScript

JavaScript je jezik za pisanje skripta koji se temelji na objektima. U našoj aplikaciji koristili smo JavaScript kako bismo dodali funkcionalnost aplikaciji. HTML i CSS koristimo samo za prikaz i uređivanje elemenata na stranici (tekst, forme, linkovi i tabele s izgledom), ali ne pruža mogućnost dinamike. Korišćenje JavaScripta nam omogućuje veću kontrolu nad web stranicom i njenim ponašanjem. Kombinacija JavaScripta, HTML-a i CSS-a čini DHTML (Dinamički HTML).

Naziva se objektno zasnovan jer programer ne samo da definiše vrstu podataka, već i vrstu operacija (funkcija) koje se mogu primijeniti na strukture podataka. Na taj način, struktura podataka postaje objekt koji sadrži podatke i funkcije. Takođe, mogu se definisati i odnosi između više objekata.

Skriptni jezik znači da se kod interpretira u izvršnom okruženju, bez potrebe za prethodnom kompilacijom, odnosno pretvaranjem u mašinski jezik s nulama i jedinicama. JavaScript se izvršava na strani klijenta, tj. na računalu na kojem je pokrenut sadržaj.

JavaScript je najpopularniji skriptni jezik na internetu i podržavaju ga svi poznati pretraživači poput Google Chrome-a, Microsoft Edge-a, Internet Explorera, Mozilla Firefox-a, Netscape-a i Opere.

## JSON

JSON (JavaScript Object Notation) je tekstualni format za prenos podataka. Izveden je iz formata za zapis objekata u JavaScript programskom jeziku, no danas se koristi u većini modernih programskih jezika. Glavna svrha JSON-a je olakšati prenos podataka, te se često koristi kao zamena za XML. Jedna od prednosti JSON-a u odnosu na XML je manji memorijski prostor koji je potreban za čuvanje istih podataka. Iako je tesno povezan s JavaScriptom, danas gotovo svi programski jezici podržavaju JSON.

JSON se temelji na dve strukture:

* Kolekcija parova ključ/vrednost. U programskim jezicima, ovaj koncept se ostvaruje putem objekata, zapisa, struktura, rečnika, lista s ključevima ili asociranih nizova.
* Uređena lista vrednosti. U većini jezika, ovo se ostvaruje putem niza, vektora, liste ili sekvence.

## React

React je najpopularnija front-end JavaScript biblioteka na polju veb razvoja. Koriste ga velike kompanije i startapi (Facebook, Netflix, Airbnb, Instagram i New York Times…). React donosi mnoštvo prednosti, čineć i ga boljim izborom od ostalih okvira poput Angular.js i drugih. Jednostavniji je I lakši za savladati početnicima od većine tih okvira. React je JavaScript biblioteka stvorena za izgradnju brzih i interaktivnih korisničkih interfejsa za veb i mobilne aplikacije. To je otvorena biblioteka, zasnovana na komponentama. Komponente su u našpoj aplikaciji realizovane u folderu components, gde svaka predstavlja fajl za sebe. U arhitekturi Controller-View-Model(MVC), sloj prikaza odgovoran je za to kako aplikacija izgleda i kako se ponaša. React je kreirao Jordan Valke, softverski inženjer na Fejsbuku.

React-ova popularnost danas je nadmašila popularnost svih ostalih front-end razvojnih okvira. Evo nekih razloga zašto:

* Lako stvaranje dinamičkih aplikacija: React olakšava stvaranje dinamičkih veb aplikacija jer zahteva manje kodiranja i nudi više funkcionalnosti, za razliku od JavaScript-a, gde kodiranje vrlo brzo postaje složeno.
* Poboljšane performanse: React koristi virtuelni DOM, čime brže stvara veb aplikacije. Virtuelni DOM upoređuje prethodna stanja komponenti i ažurira samo stavke u stvarnom DOM-u koje su promenjene, umesto da ponovo ažurira sve komponente, kao što to rade konvencionalne veb aplikacije, što ga čini znatno bržim.
* Komponente za višekratnu upotrebu: Komponente su gradivni elementi bilo koje React aplikacije, a jedna aplikacija se obično sastoji od više komponenti. Ove komponente imaju svoju logiku i kontrolu i mogu se ponovo koristiti tokom aplikacije, što zauzvrat dramatično smanjuje vreme razvoja aplikacije, I smanjuje količinu koda (na sličan način kao što se to postiže funkcijama)
* Jednosmerni tok podataka: React prati jednosmerni tok podataka. To znači da prilikom dizajniranja aplikacije React programeri često podređuju decu komponente roditeljskim komponentama. Buduć i da podaci teku u jednom pravcu, samo od roditelja ka deci, postaje lakše otklanjati greške i znati gde se problem javlja u aplikaciji u datom trenutku.
* Mala kriva učenja: React je lako naučiti, jer uglavnom kombinuje osnovne HTML i JavaScript koncepte sa nekim korisnim dodacima.
* Može se koristiti za razvoj veb i mobilnih aplikacija: Već znamo da se React koristi za razvoj veb aplikacija, ali to nije sve što može. Postoji okvir nazvan React Native, izveden iz samog React-a, koji je izuzetno popularan i koristi se za stvaranje prelepih mobilnih aplikacija. Dakle, u stvarnosti, React se može koristiti za izradu veb i mobilnih aplikacija.
* Namenski alati za lako otklanjanje grešaka: Facebook je objavio Chrome dodatak koji se može koristiti za otklanjanje grešaka u React aplikacijama. Ovo čini proces otklanjanja grešaka u veb aplikacijama bržim i lakšim. Gore navedeni razlozi više nego opravdavaju popularnost biblioteke React i zašto je prihvata veliki broj organizacija i preduzeća.

## Laravel

Laravel je dobro poznati, besplatni PHP veb radni okvir otvorenog koda, kojeg je osmislio i kreirao Tejlor Otvel. Od svog početka, softver je bio namenjen razvoju veb aplikacija, posebno onih koje prate MVC arhitekturu zasnovanu na Symfony-ju ( takođe php radni okvir za veb aplikacije). Takođe, Laravel nudi pristup odnosnim bazama podataka na različite načine, i koristi osobine koje pomažu u održavanju i primeni aplikacija. Definiše međusobno povezane modele, kontrolere (koji definišu metode za upravljanje podacima u bazi, gde nije neophodno uneti tačan SQL upit, koji se krije iza ovih metoda), resurse(koji služe za prikaz podataka korisniku). Ovim podacima se može pristupiti unosom rute, I slanjem određenog HTTP zahteva(POST, GET, PUT, DELETE) u aplikaciju postman, ili pomoću browsera. Jednostavno rečeno, Laravel je veb radni okvir koji pomaže pri izgradnji novih aplikacija za veb, uključujući i stvari poput API-ja ili veb usluga.

## MVC Pattern

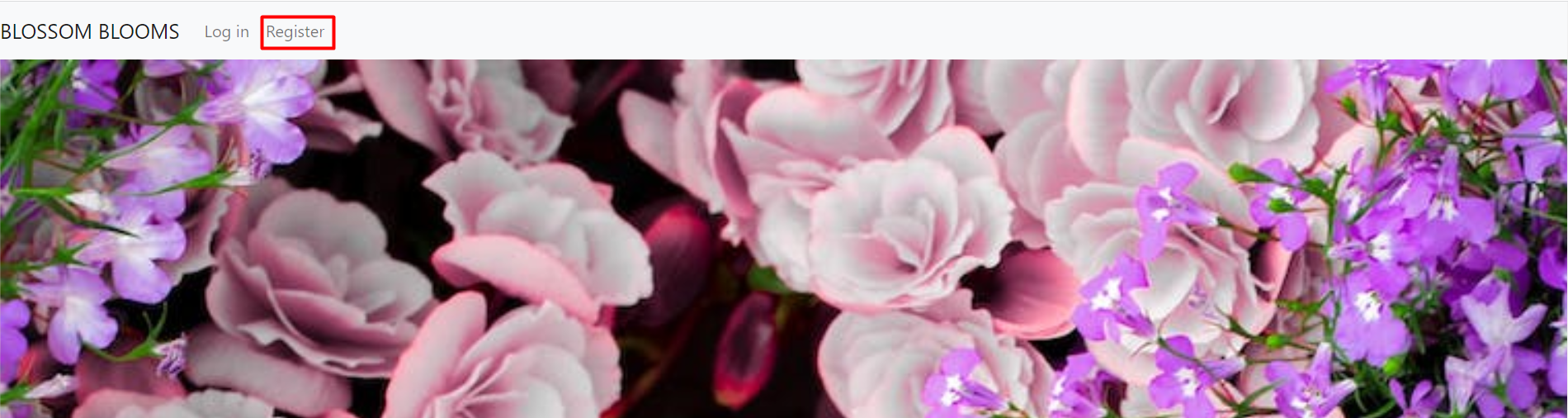
U ovoj aplikaciji korišćen je MVC Pattern. Okvir Model-View-Controller (MVC) je arhitektonski obrazac koji odvaja aplikaciju u tri glavne logičke komponente Model, View i Controller. MVC razdvaja poslovnu logiku i sloj prezentacije jedan od drugog. Tradicionalno se koristio za desktop grafički korisnički interfejs (GUI). Controller I Model su realizovani u Laravelu, dok je View vezan za React. MVC pruža front i back-end za bazu podataka, korisnika i komponente za obradu podataka. Razdvajanje softverskih sistema na front i back deo pojednostavljuje razvoj i razdvaja održavanje.

# Korisničko uputstvo

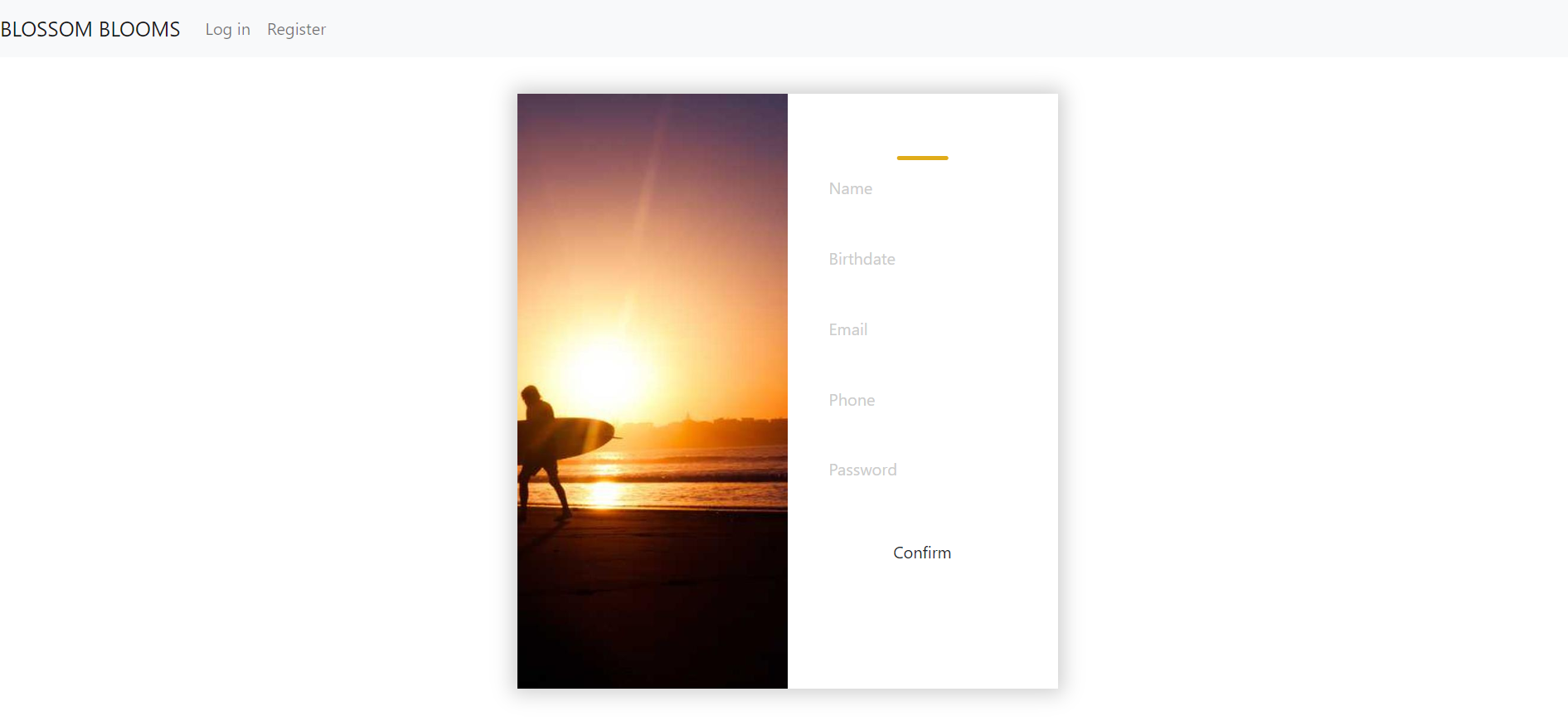
Predstavićemo prethodno navedene slučajeve korišćenje u cilju efikasnog i uspešnog snalaženja korisnika na našem veb sajtu.

1. Register

Ukoliko nema nalog, korisnik klikom na dugme Register koje se nalazi na navigacionom meniju pristupa stranici za registraciju koja će biti prikazana u par narednih slika.

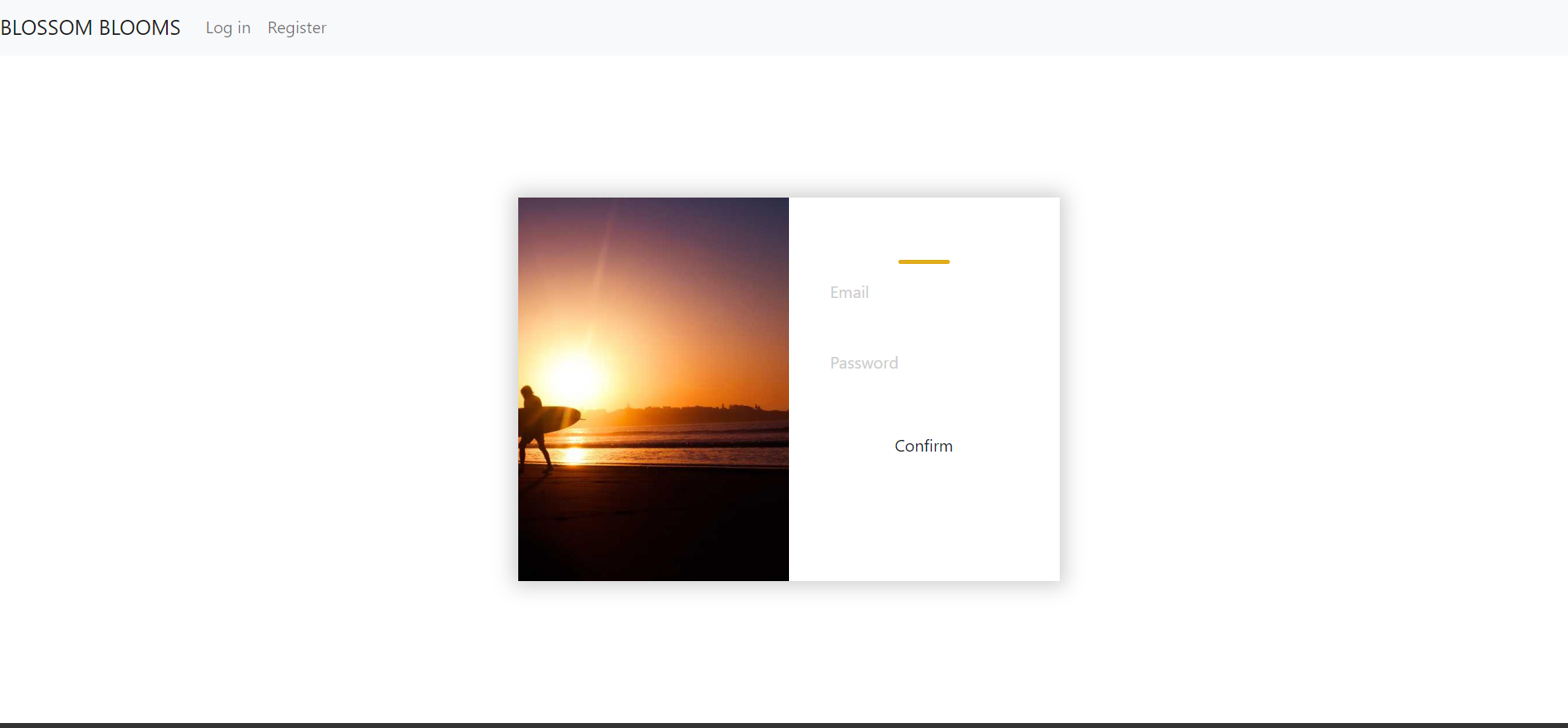


Nakon što svoje podatke ukuca u polja, pritisne dugme Confirm koje će ga odvesti na Login stranicu.



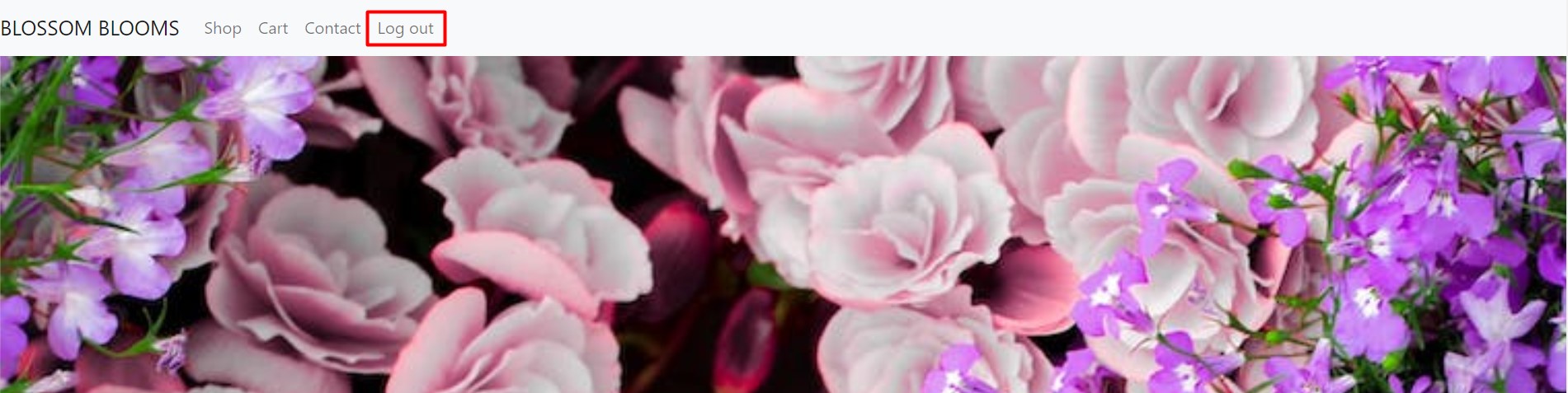
1. Login

Nakon što se registrovao, kao što smo i naveli iznad, prelazi na stranicu za Login gde će ukucati svoje kredencijale za pristup sajtu. Nakon ovoga, korisnik se prebacuje na Shop stranu.



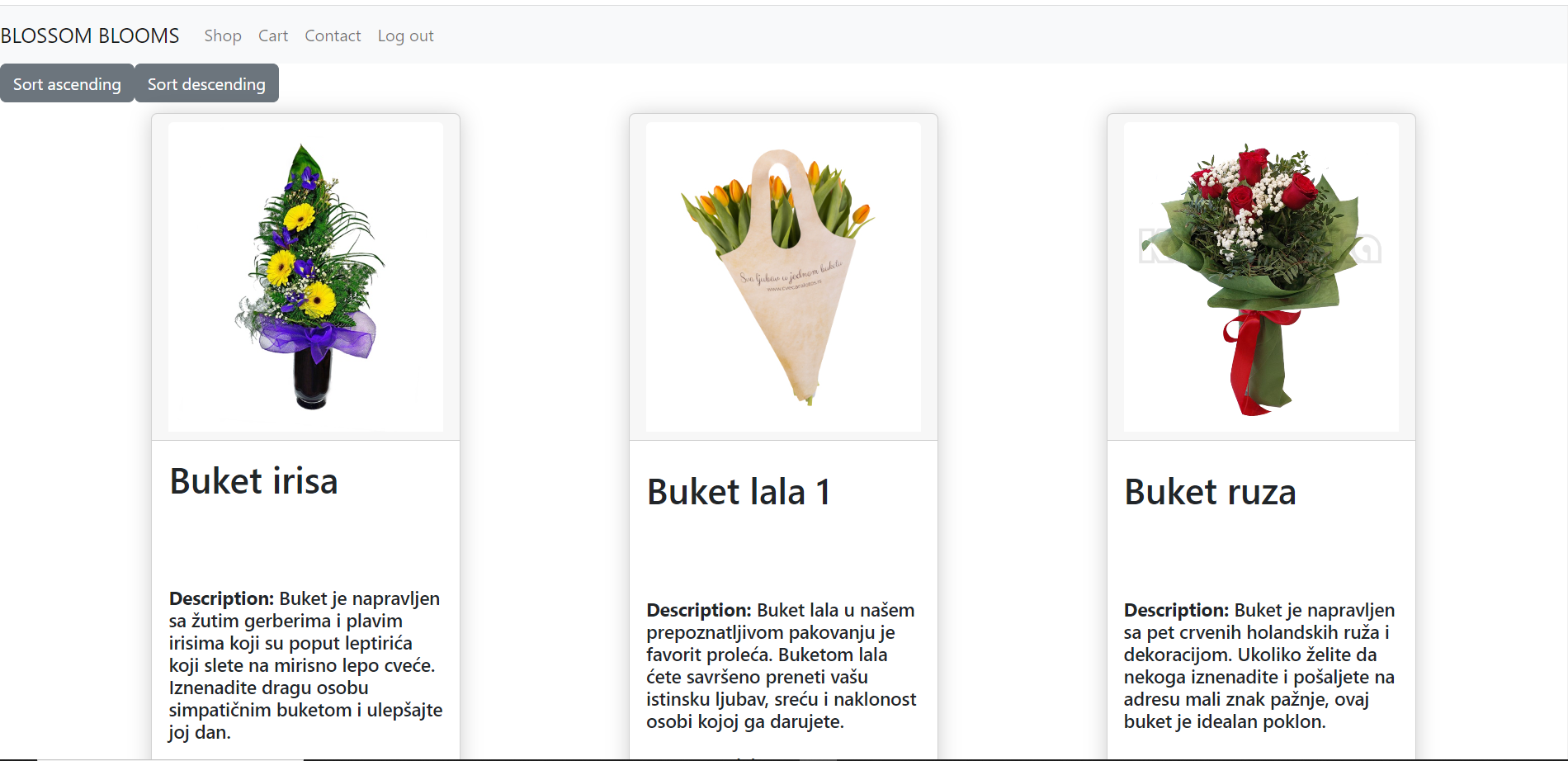
1. Logout

Ukoliko korisnik želi da se odjavi sa svog profila na sajtu, to može učiniti tako što će na mestu gde je stajalo Log in, pritisnuti dugme Log out, čime će se uspešno izlogovati sa svog profila i vratiti na početnu stranu.



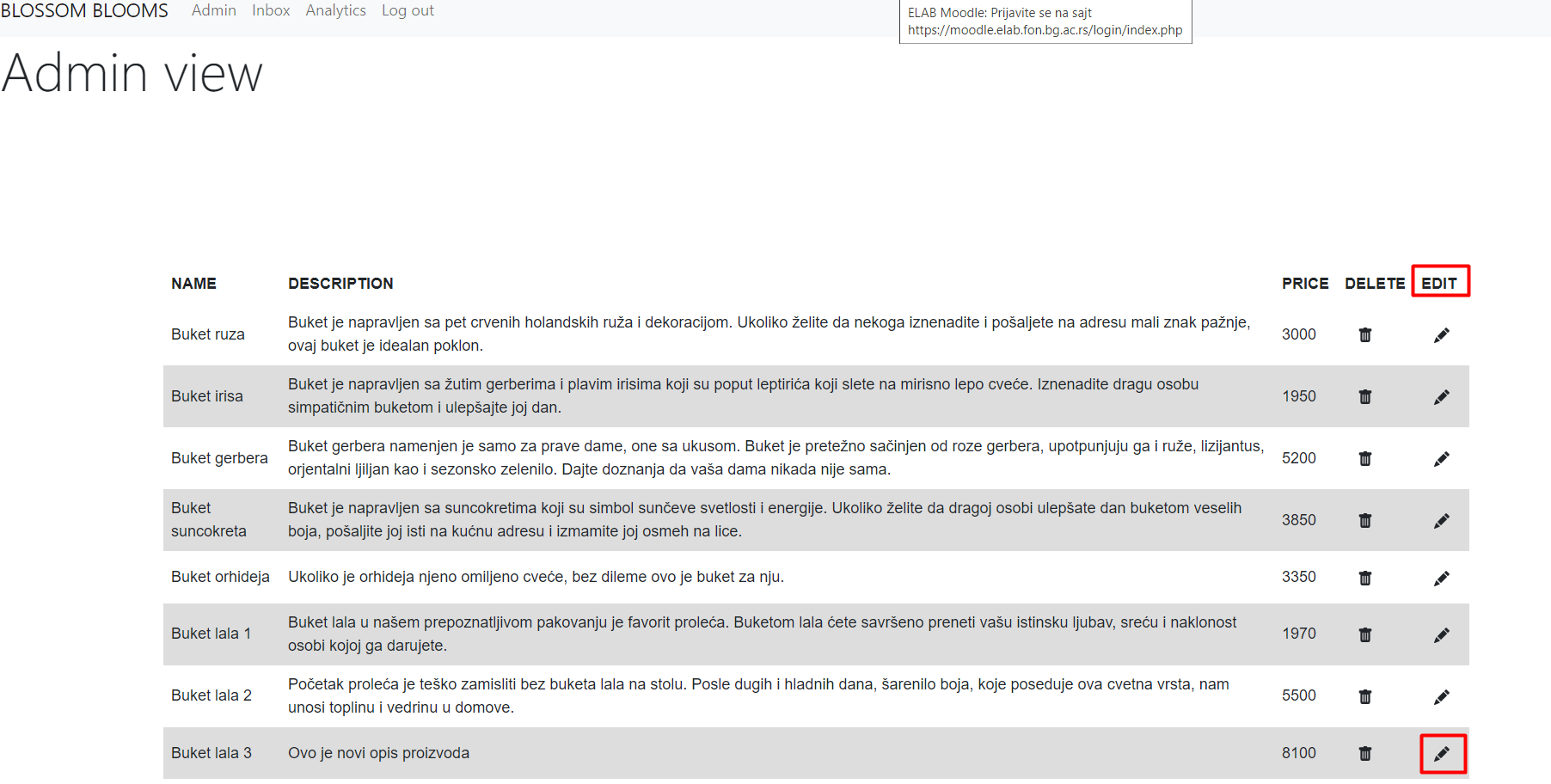
1. Prikaz cveća (buketa)

Trenutka kada se korisnik uloguje na prodavnicu, biće prebačen na Shop stranici, gde se nalaze svi proizvodi koji su u ponudi odnosno svi buketi koje može da naruči, kao i sve informacije vezane za iste- naziv, opis, cena i slika. Omogućeno je i sortiranje buketa po opadajućoj ili rastućoj ceni buketa.

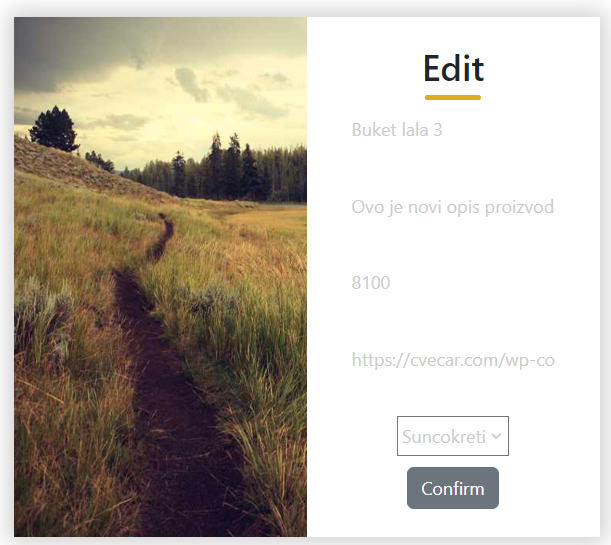


1. Izmena cveta(buketa)

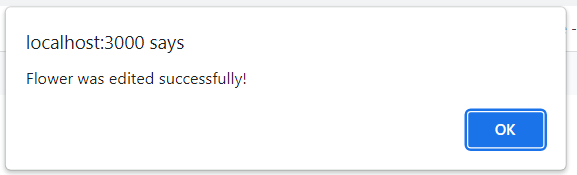
Ovaj slučaj korišćenja vezan je za admina koji može da upravlja asortimanom cvećare time što će dodati, obrisati ili izmeniti aranžman koji je u ponudi. Dakle, kada se admin uloguje sa svojim kredencijalima, izgled njegovog Dashboarda će biti ovakav:



On bira opciju edit za buket koji želi da izmeni i samim tim će mu se prikazati forma za izmenu.

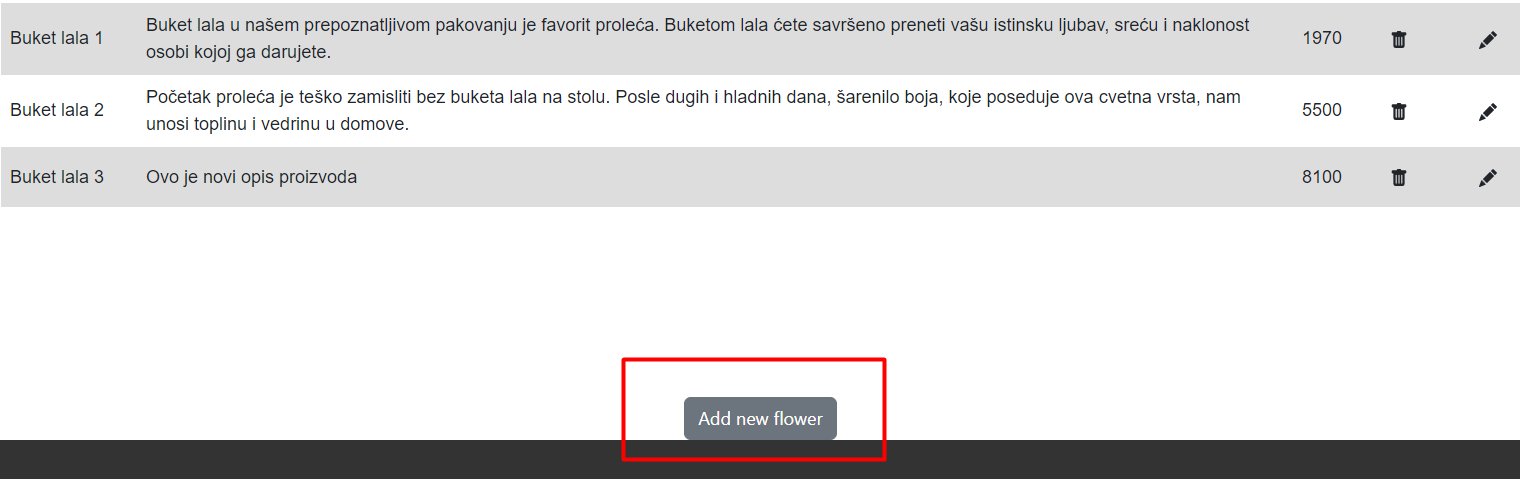


Kao što vidimo, polja su već popunjena trenutnim podacima za dati aranžman, tako da korisnik unosi željene izmene i klikom na dugme Confirm, on potvrđuje promenu, nakon čega će izaći poruka o uspešnosti izmene.

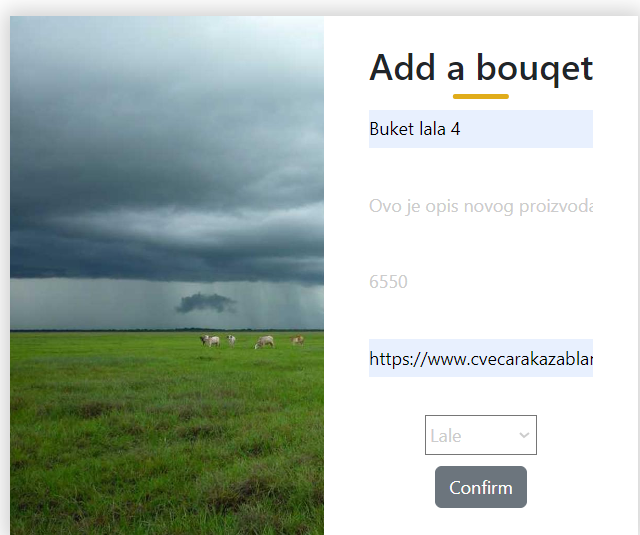


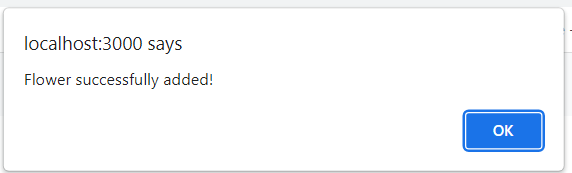
1. Dodavanje cveta(buketa)

Na samom dnu, ispod tabele nalazi se dugme Add new flower, na koje admin klikne ukoliko želi da doda novi aranžman, što ga vodi na formu za dodavanje buketa.

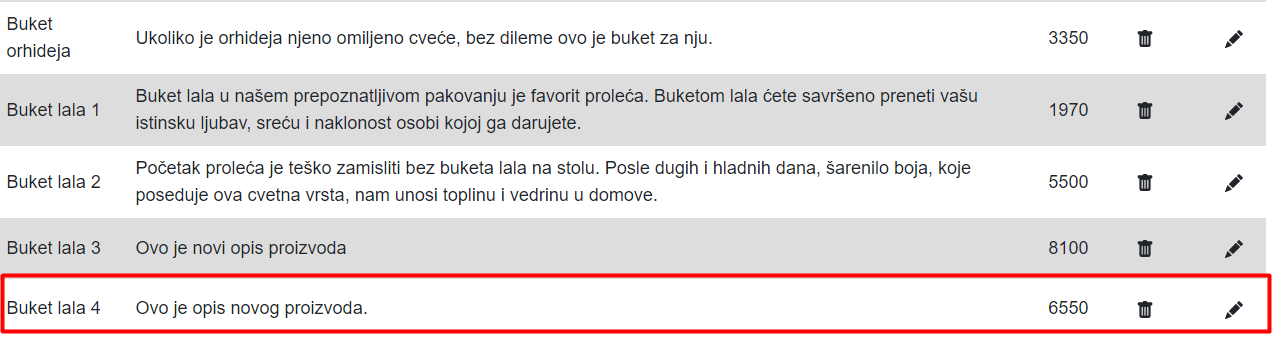


Unosi sve potrebne podatke za buket i pritiska dugme Confirm nakon čega se buket čuva u bazi uz poruku da je sačuvan.



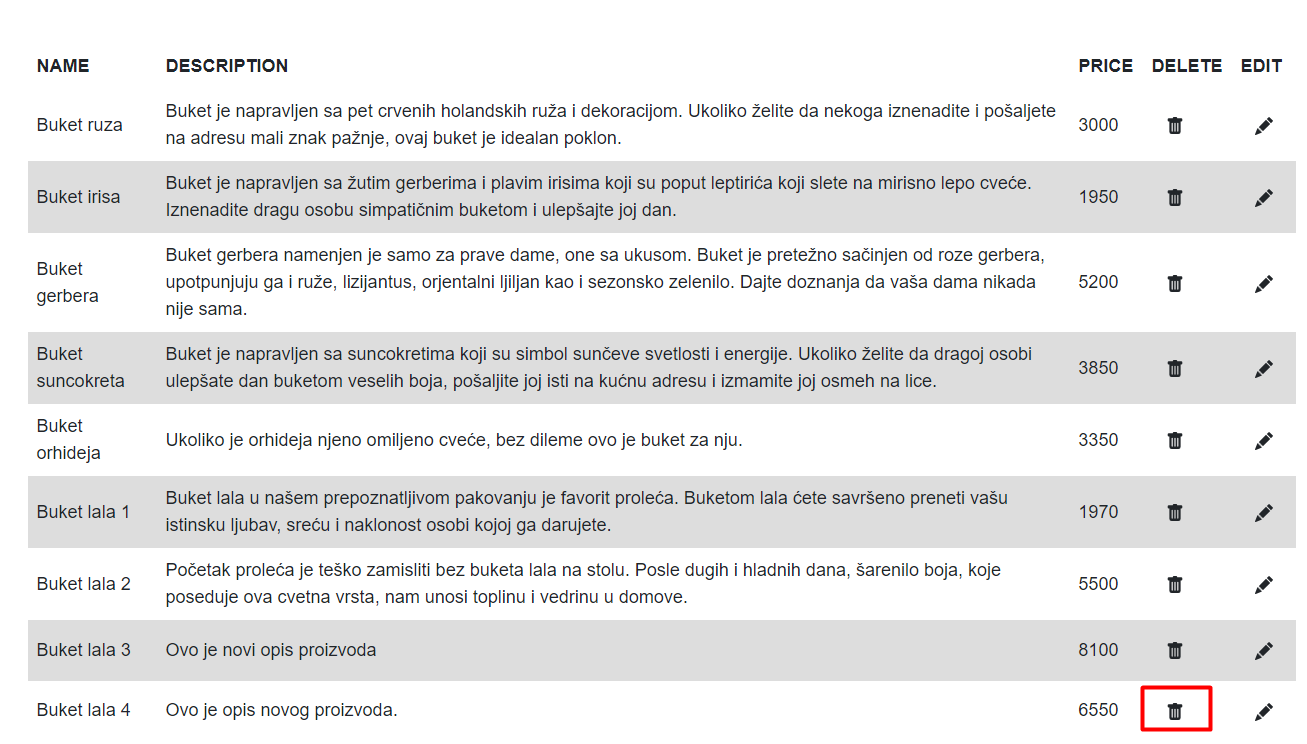


Možemo videti da je nov buket dodat u tabelu.

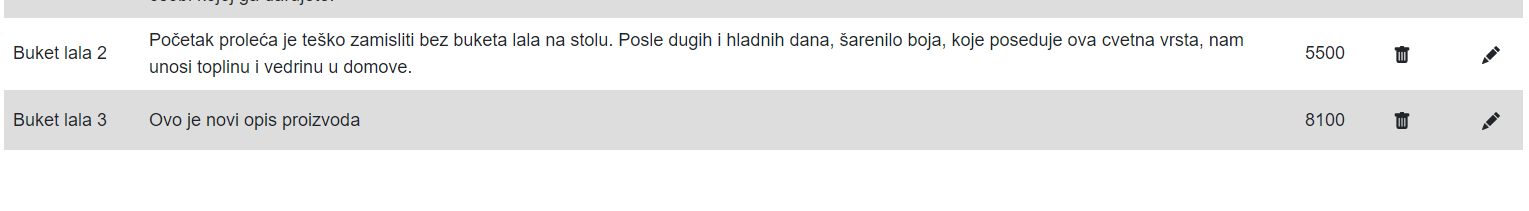


1. Brisanje cveta(buketa)

Ono što admin radi jeste da u delu gde piše Delete klikne na ikonicu kod određenog buketa koji želi da obriše.

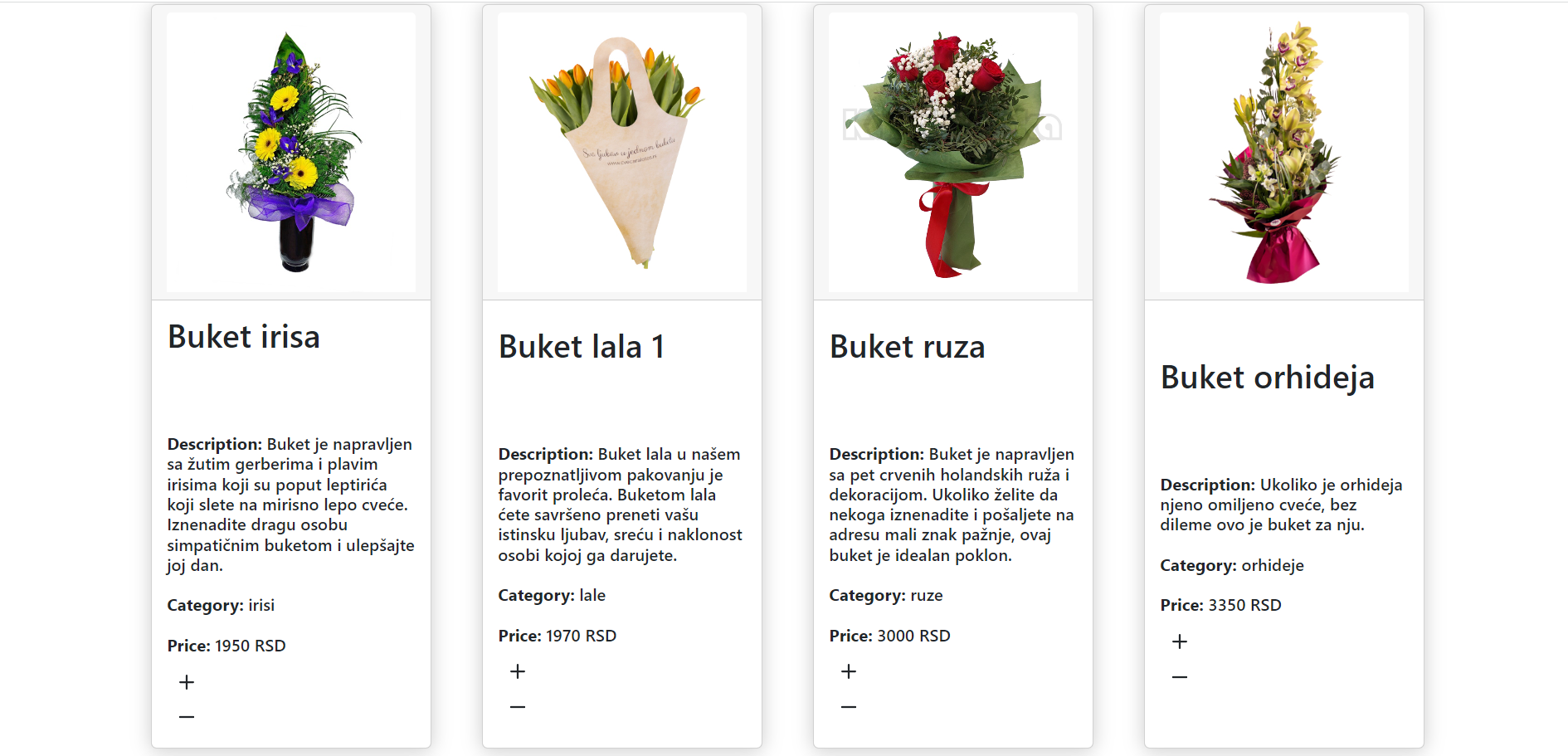


Vidimo na narednoj slici da je proizvod zaista obrisan.

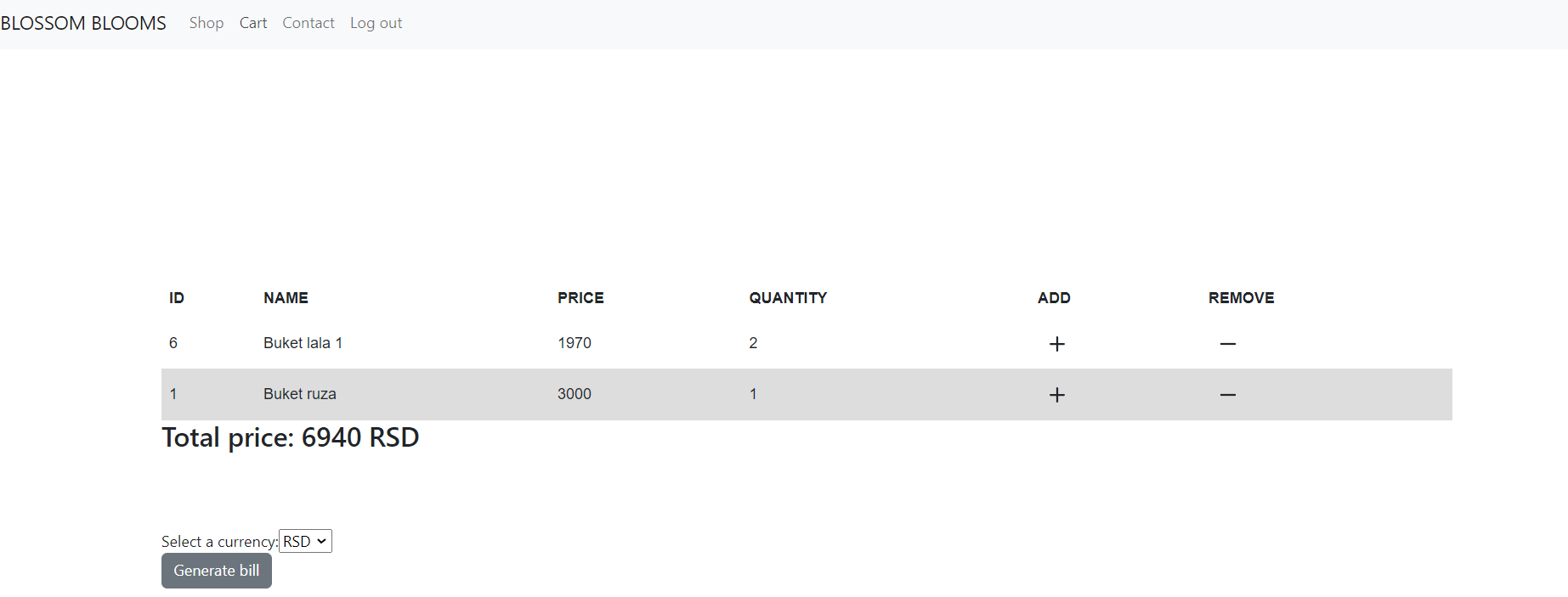


1. Dodavanje proizvoda u korpu

Korisnik će klikom na + dodati buket u korpu. Ukoliko ipak ne želi konkretno taj buket, uvek ga može klikom na – izbaciti iz korpe.



Sve dodate bukete, on može videti tako što će kliknuti na Cart, što će ga odvesti pravno na stranicu korpe gde će imati pregled svoje narudžbine, kao i ukupan iznos za plaćanje.

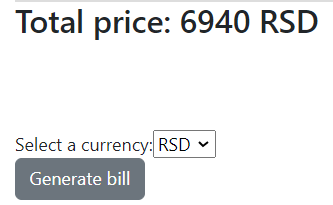


1. Brisanje proizvoda iz korpe

Ukoliko se korisnik ipak predomisli i želi da smanji količinu naručenih buketa, pa čak i u potpunosti ukloni iz svoje narudžbine, sve što je potrebno je da u delu Remove on pritisne – kod izabranog buketa koji želi da ukloni.

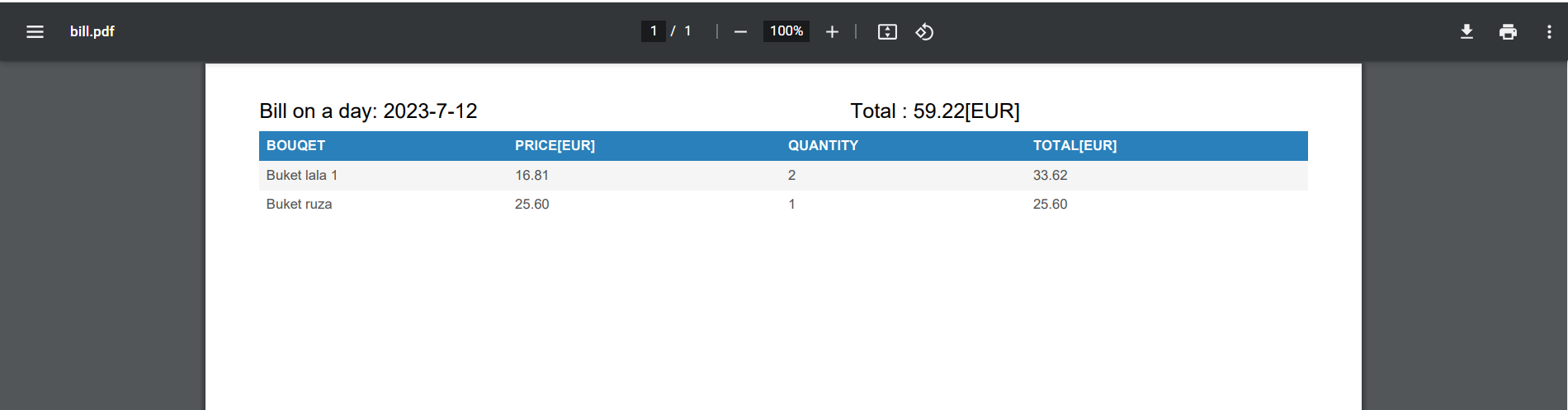
1. Izbor valute

Nakon što je završio sa odabirom, korisnik bira valutu u kojoj želi da dobije račun. To može biti u dinarima, dolarima ili evrima, što se nalazi u combo box-u pored ,,Select a currency’’.



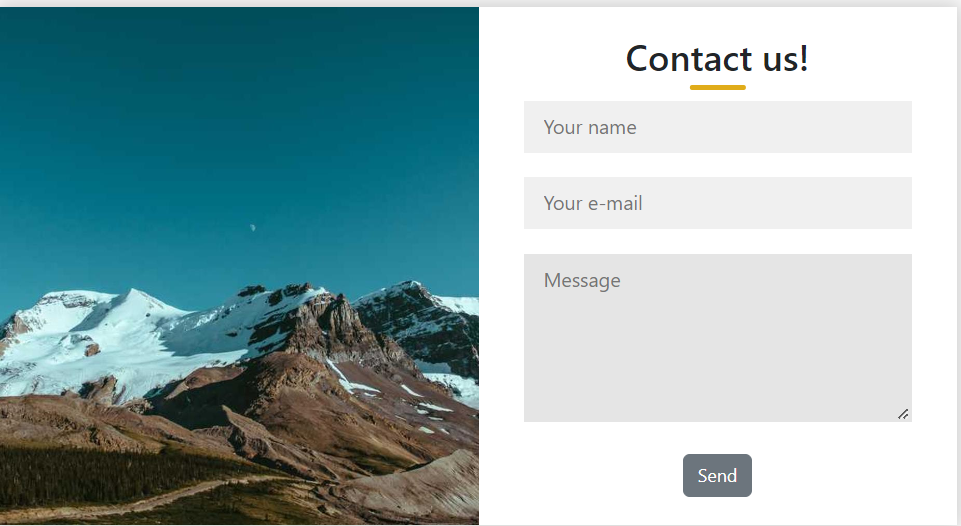
1. Generisanje računa

Nakon odabrane valute, račun se jednostavno generiše klikom na dugme Generate bill, nakon čega će korisnik dobiti svoj račun u pdf-u sa svim poručenim buketima i cenom izraženom u željenoj valuti.



1. Slanje poruke

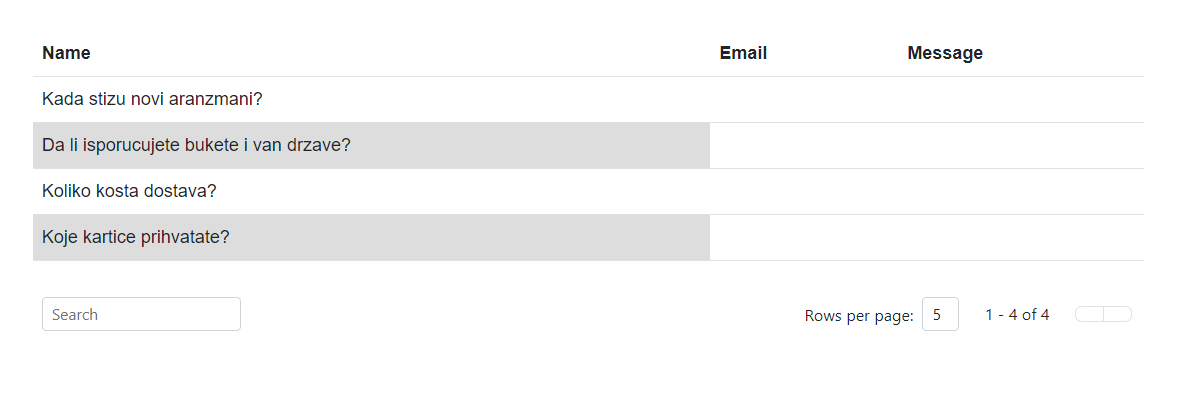
Slanje poruke se obavlja odlaskom na stranicu Contact, gde se nalazi forma koju korisnik popunjava svojim podacima-ime, email, kao i sadžaj poruke. Klikom na dugme Send, on šalje svoju poruku



1. Pregled poruka

Admin pristigle poruke može pogledati na stranici Inbox. Tamo se nalazi svaka poruka koju su korisnici poslali, zajedno sa njihovim email-om i imenom.



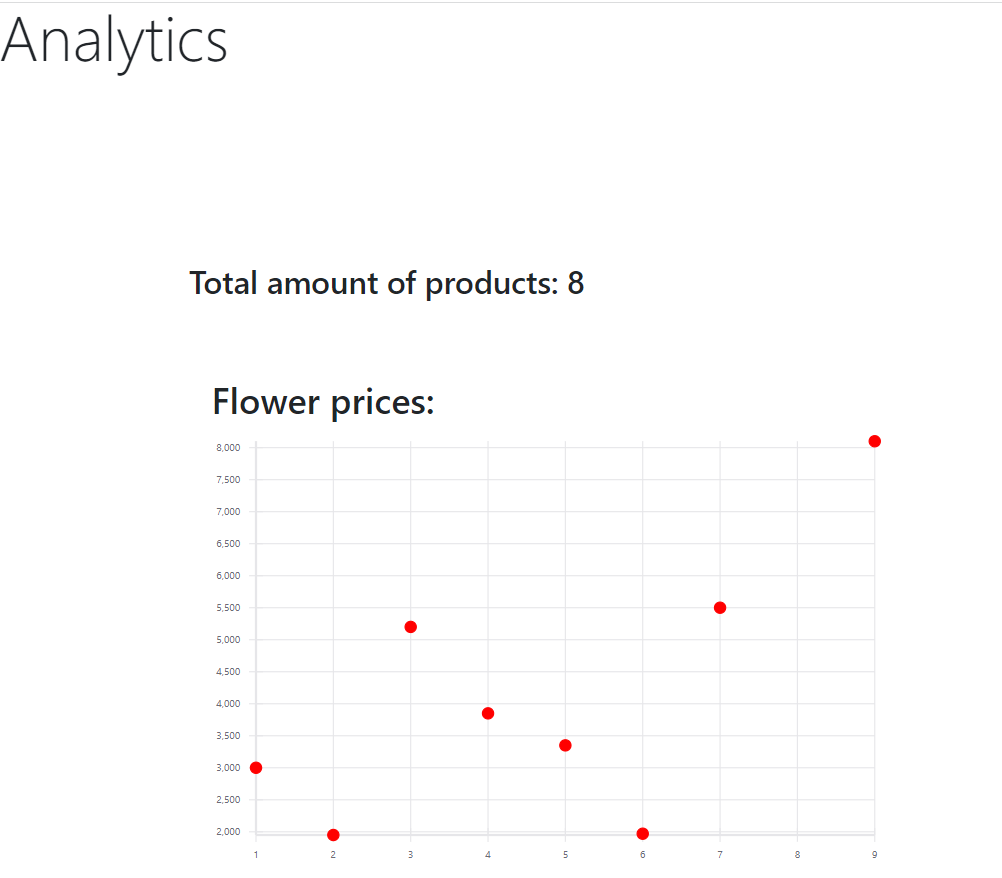


1. Pregled statistike-analitika

Admin pristupa stranici za analitiku klikom na dugme Analytics na svom navigacionom meniju.



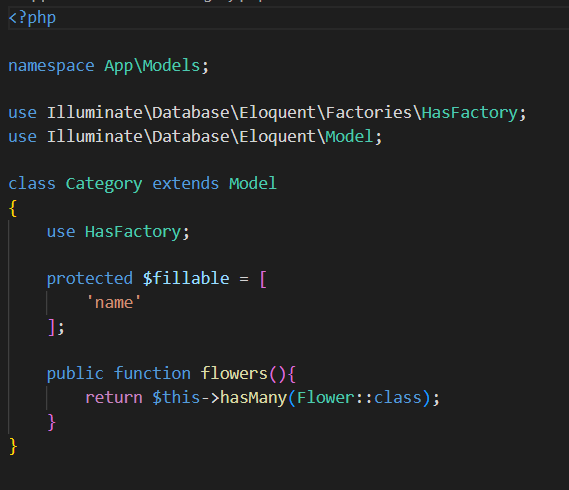
Na stranici se može videti ukupan broj proizvoda(buketa) koji su u ponudi naše cvećare. Takođe grafički je prikazana cena svakog proizvoda.



# Prikaz reprezentativnih delova koda

Pomoću Laravela implementirani su sledeći modeli: User, Category, Flower, Order, OrderItem i Message.

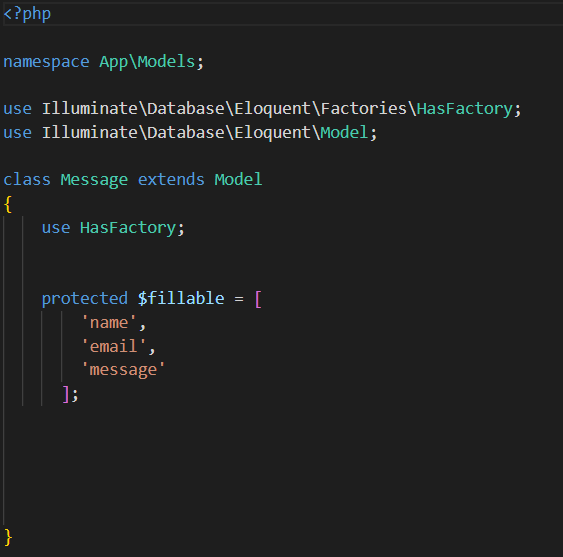
Category:



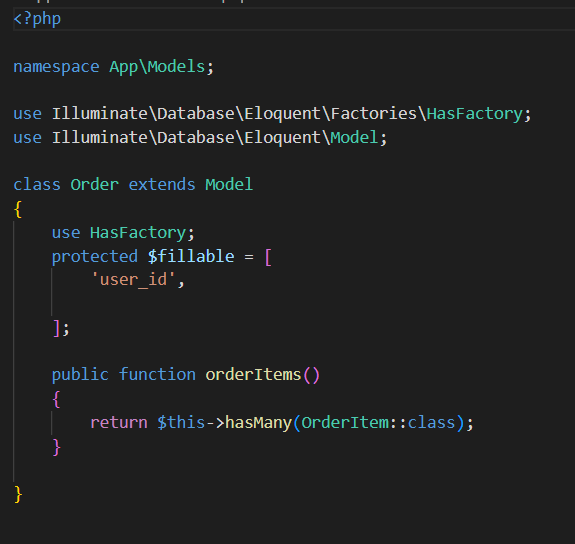
Flower:



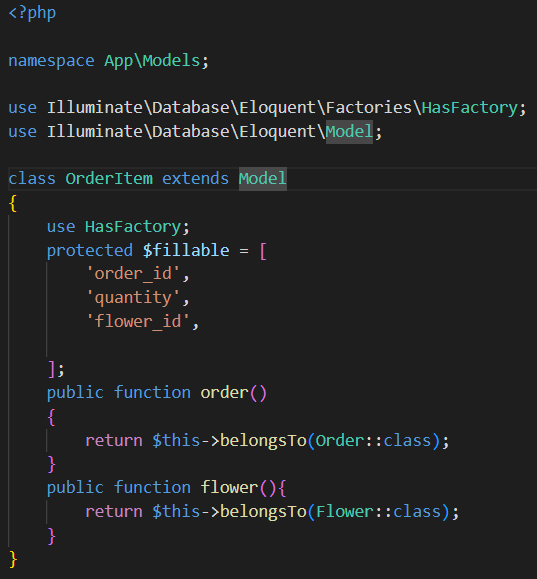
Message:



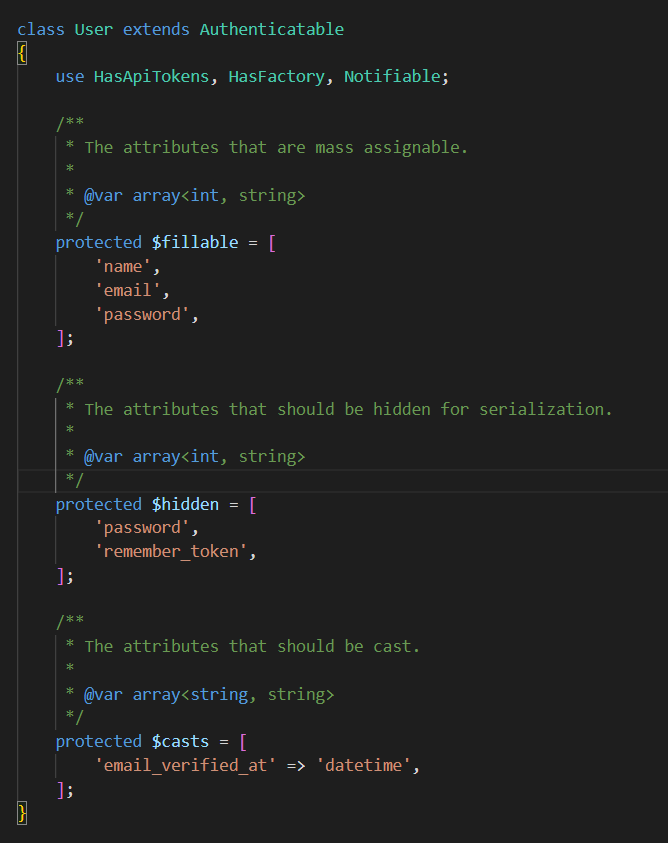
Order:



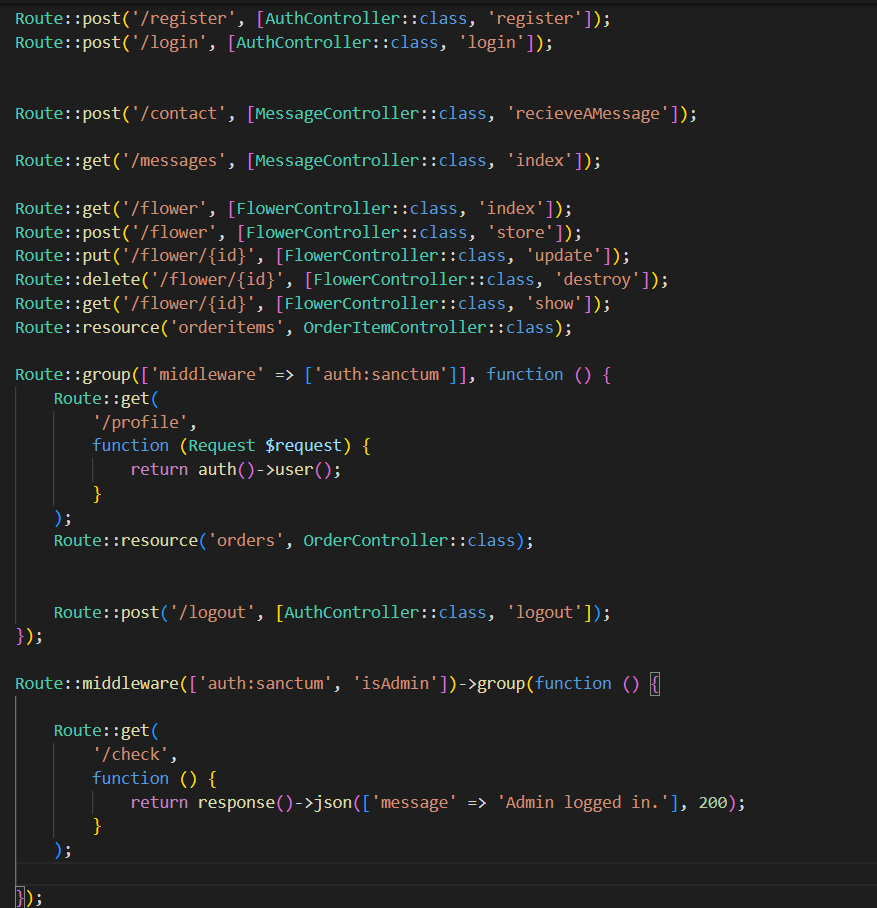
Order Item:



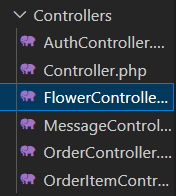
User:



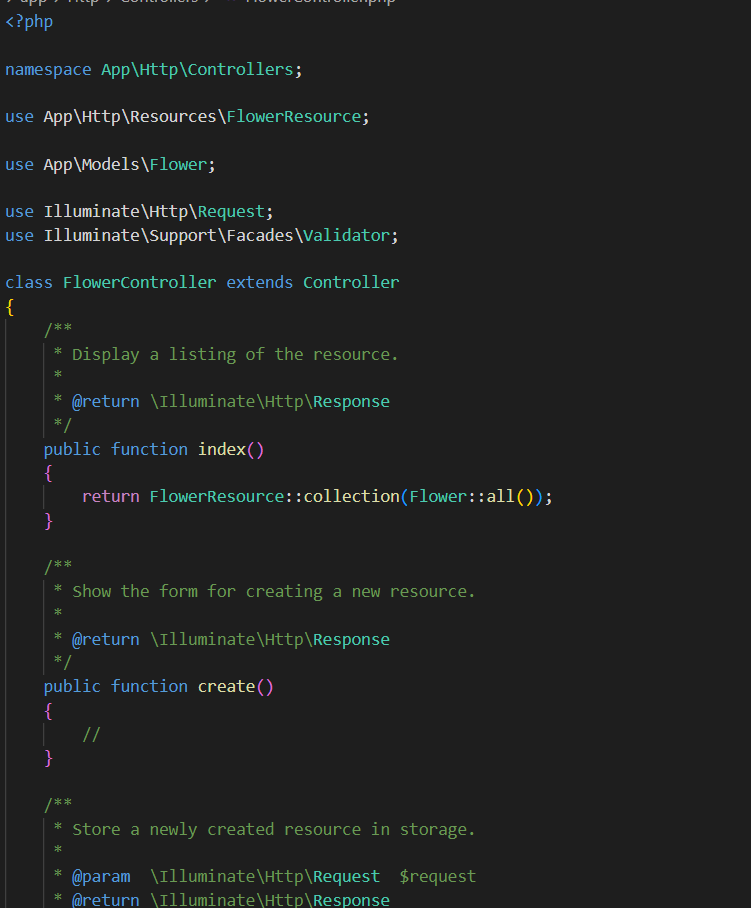
Kreirali smo sledeće rute u api.php:

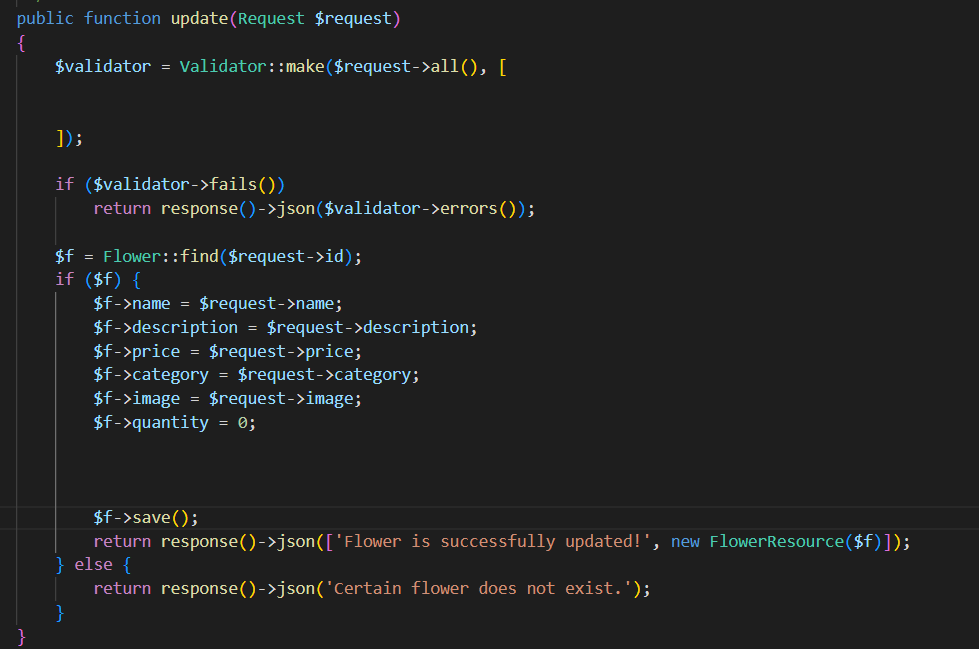


Spisak svih kreiranih kontrolera:

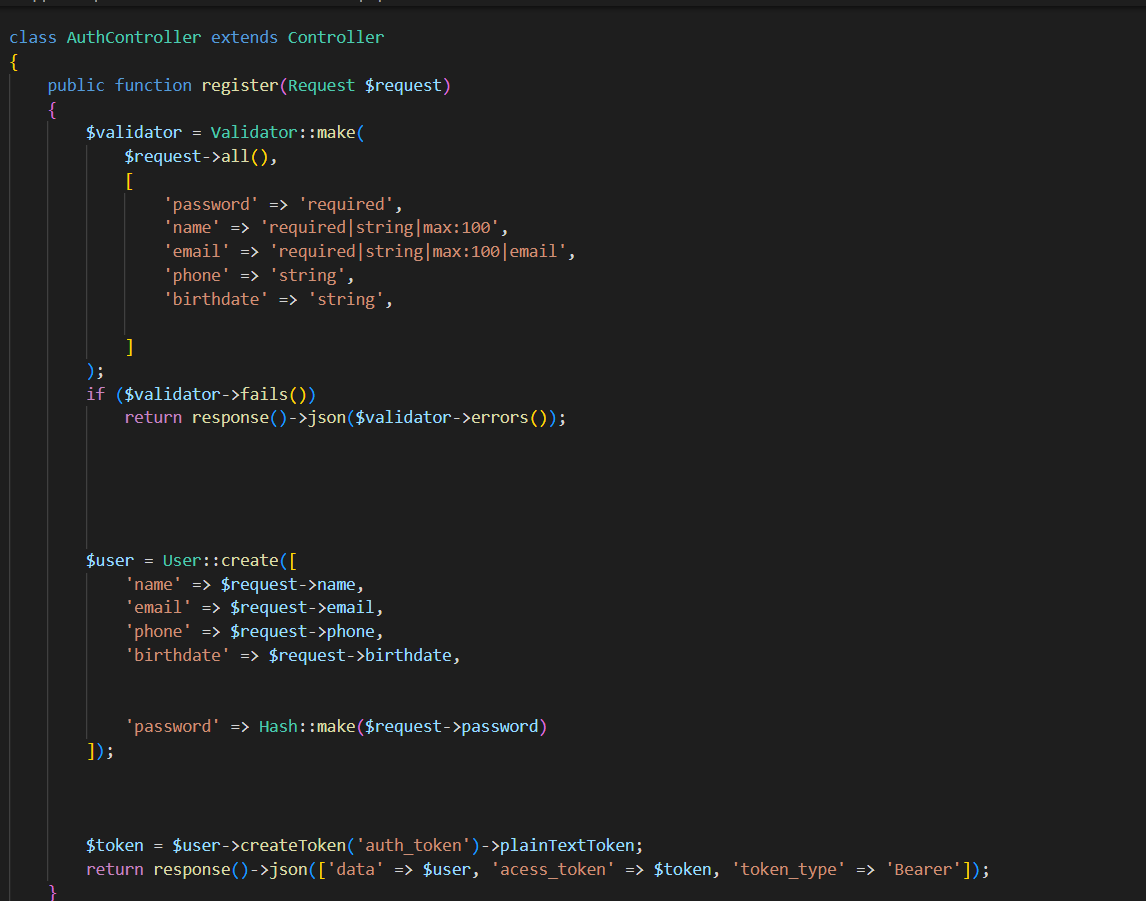


FlowerController u kom su funkcije store, update, delete i ostale:

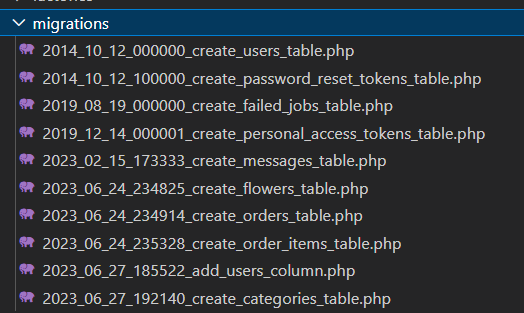




AuthController sa register, login i logout metodama.

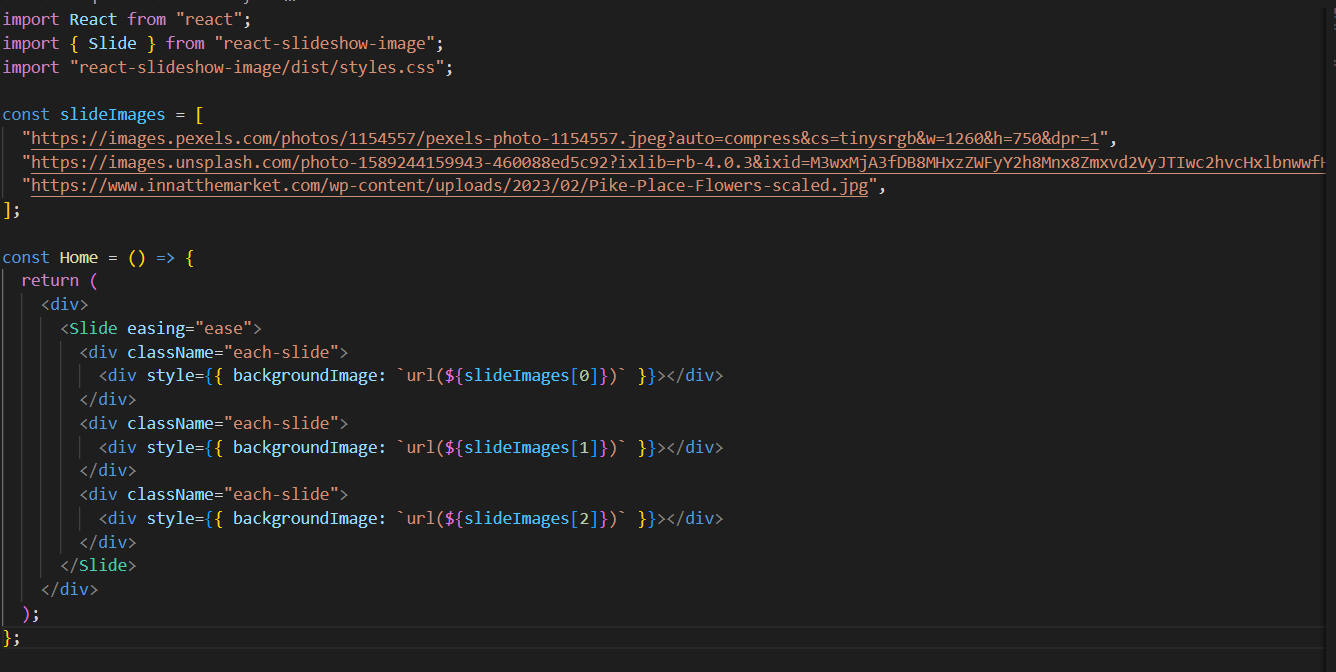


Lista migracija za kreiranje tabela flowers, categories, orders, orderitems, users.



Frontend

Početna strana- sadrži importovan Slide pomoću kog se slike smenjuju na početnoj stranici.



Footer- sadrži tekst o cvećari kao i brze linkove za pristup određenim stranicama, ruiranje obezbeđeno putem Link komponente.



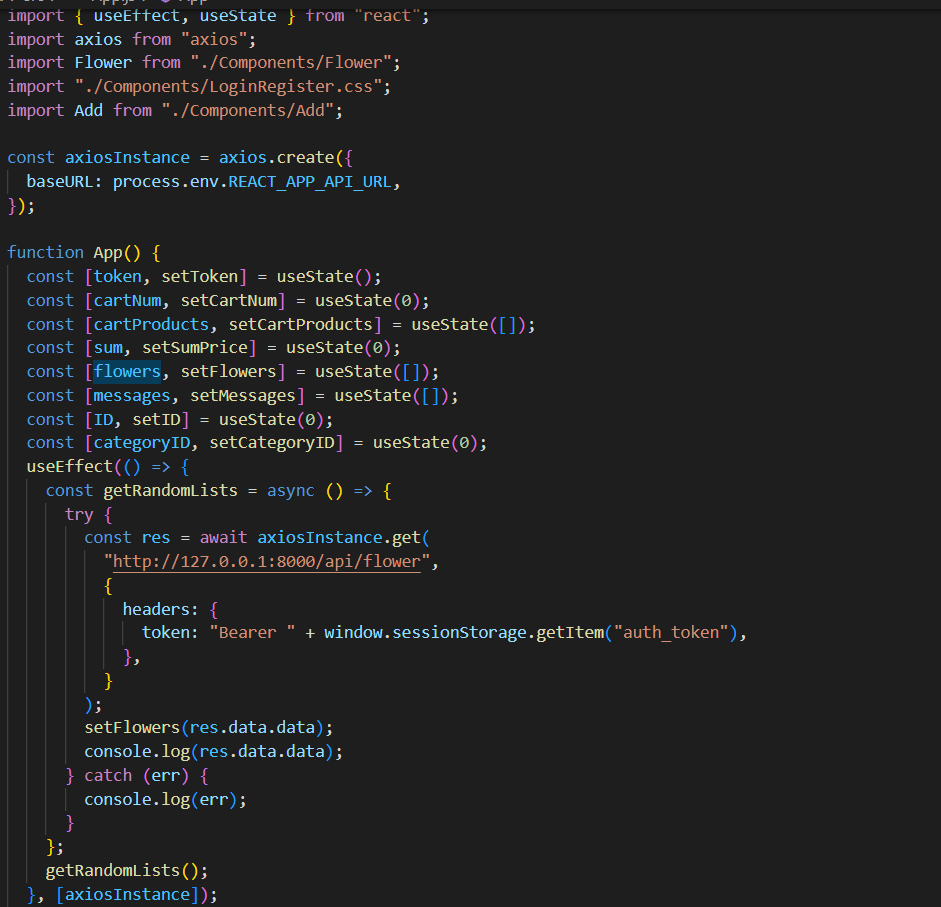
Login stranica-sadrži implementirane funkcije handleInput koje skupljaju podatke iz tekstualnih polja i zatim klikom na dugme, poziva se implementirana funkcija handleLogin koja pomoću našeg API-ja uzima podatke i ukoliko je uloga admin, vodi admina na njegovu stranicu, a običnog korisnika vodi na stranicu sa buketima.



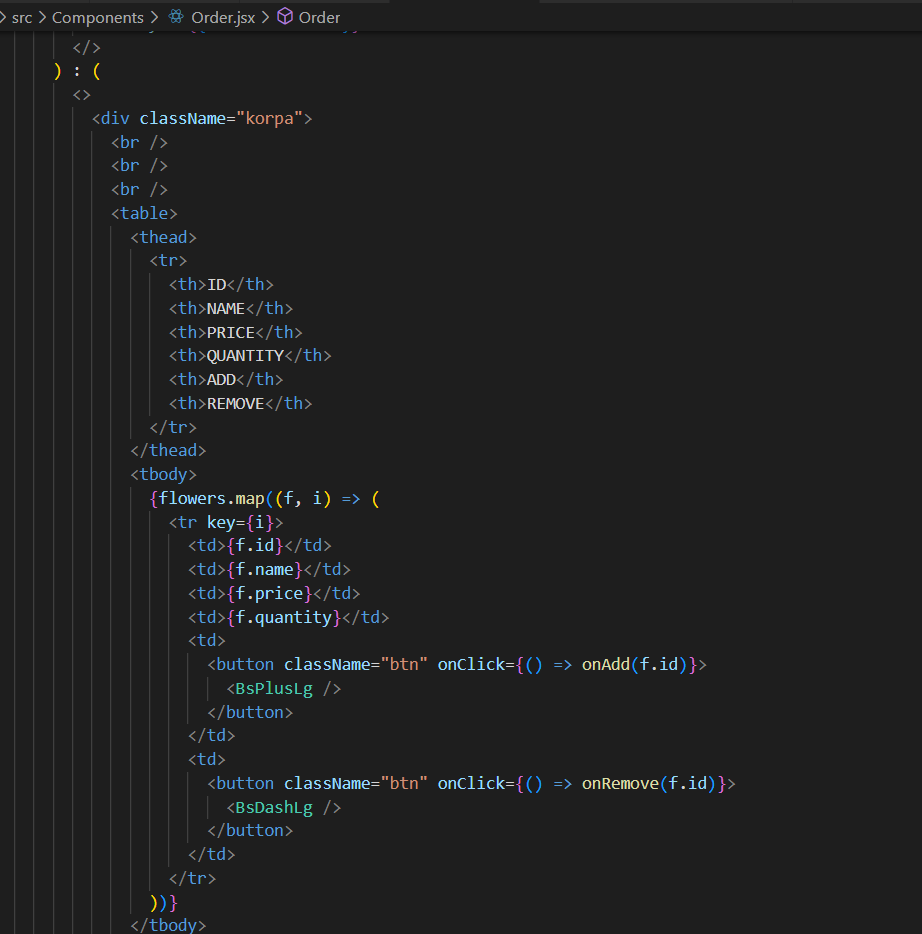
Na stranici Flower smo omogućili prikaz buketa i ikonica za dodavanje i izbacivanje iz korpe, a takođe smo i obezbedili sortiranje u opadajućem ili rastućem nizu u odnosu na cenu buketa, uz pomoć funkcije sort. Takođe importovan je FlowerItem koji predstavlja karticu sa podacima o buketu. U App.js -u dodali konstantu koja preuzima sve bukete iz baze uz pomoć rute za buket i funkcije GET.



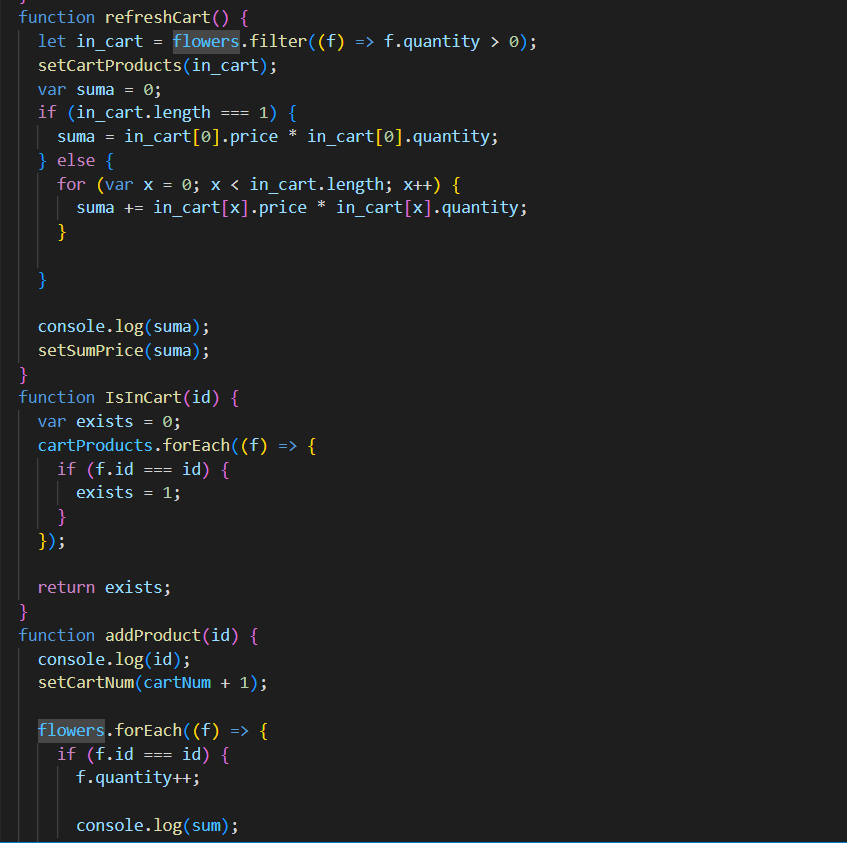




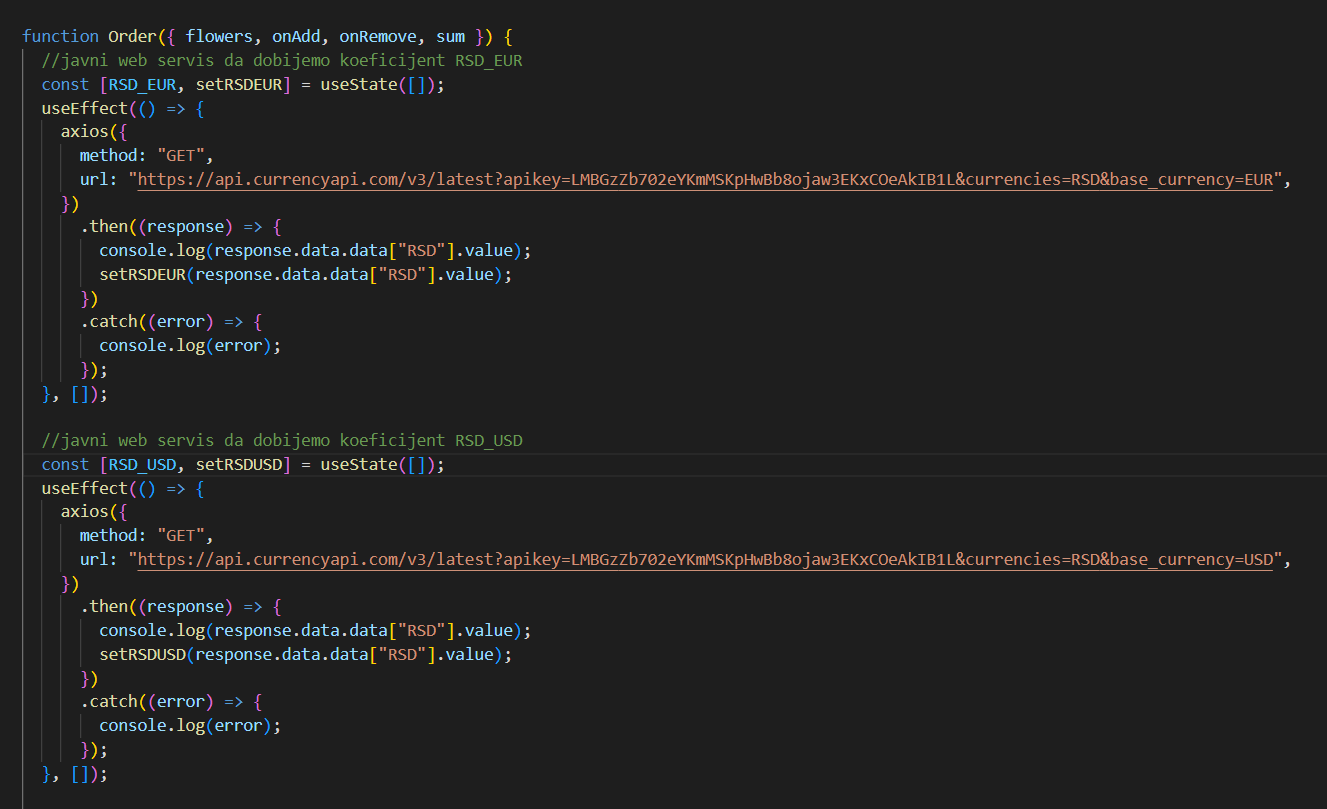
U komponenti Order, nalaze se svi proizvodi koje je korisnik izabrao na Shop Stranici, dok su funkcije za osvežavanje korpe, dodavanje u korpu i brisanje iz iste implementirane u App.js.



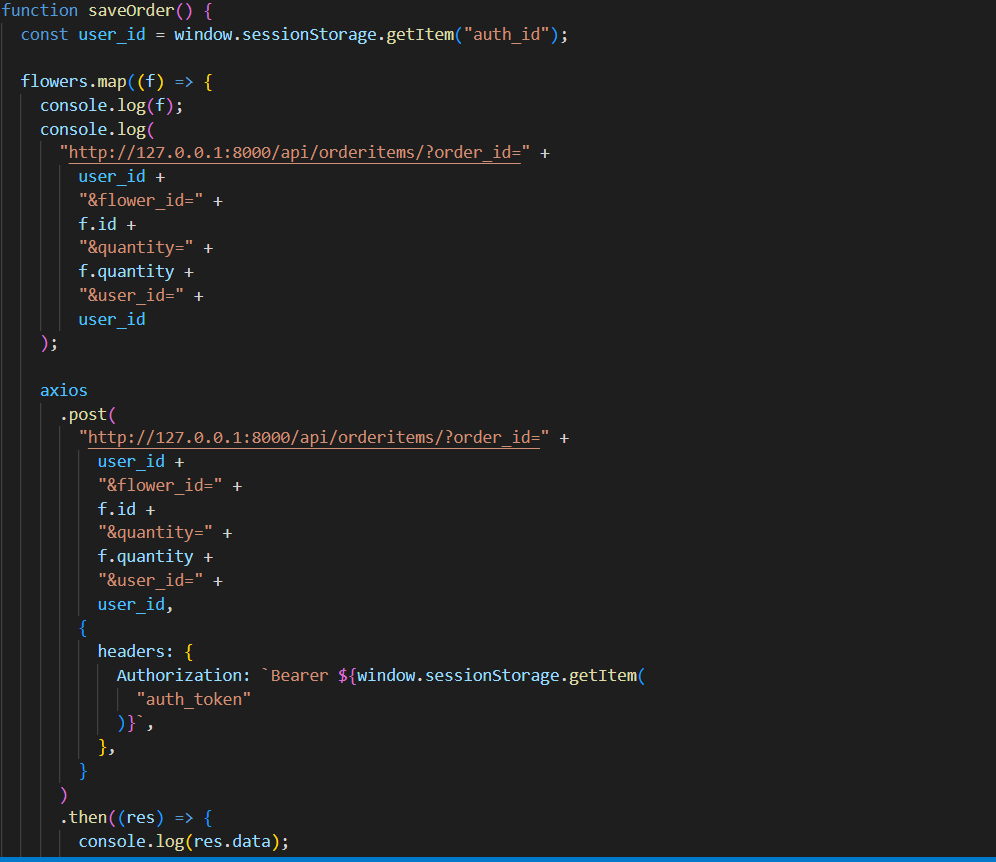
Uz pomoć ovih funkcija i useState hooks, korpa se automatski ažurira prilikom izbora proizvoda.



U komponenti Order se takođe nalazi javni API koji cenu u dinarima izražava u dolarima ili evrima i ujedno generiše i račun u PDF-u.



Takođe sadrži u funkciju koja će pomoću našeg API-ja i GET i kasnije POST metode sačuvati porudžbine u našu bazu.





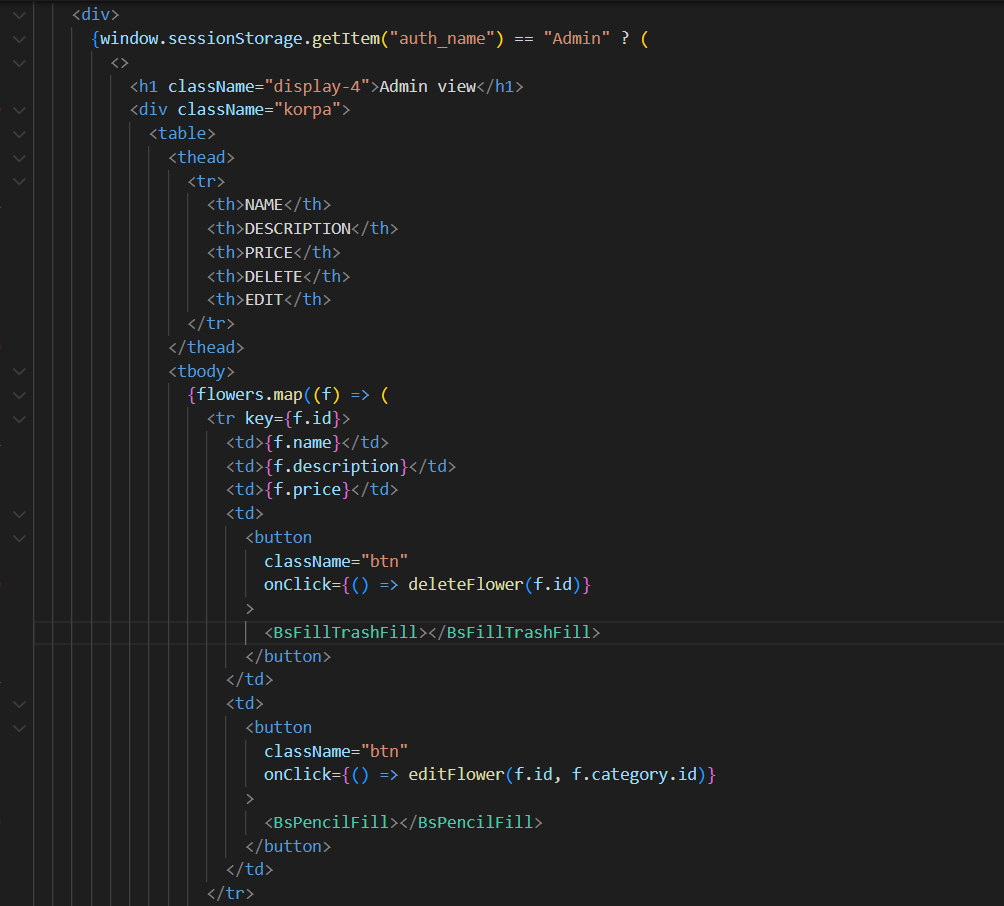
Contact stranica sadrži Google Maps API kojim se prikazuju lokacije naših cvećara na Google Maps.

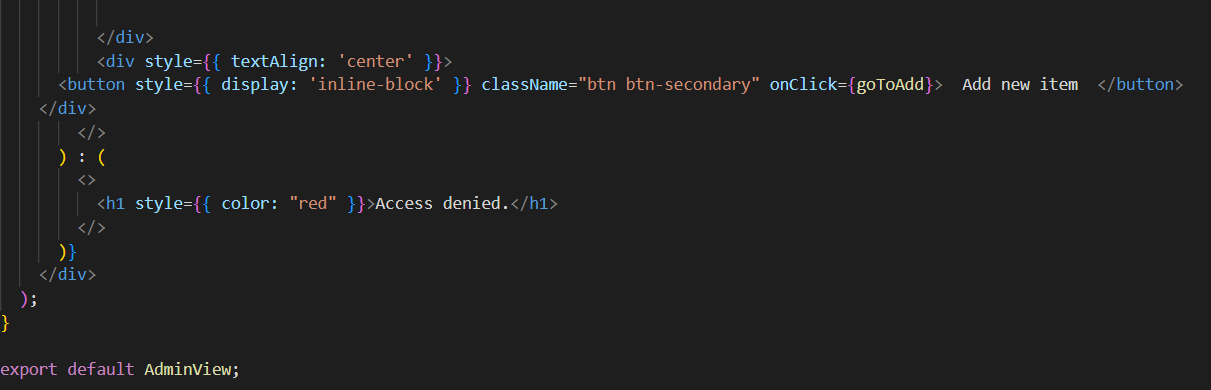


Komponenta Inbox sadrži sve poruke prikupljene sa Contact stranice i to uz pomoć komponente MDBDataTableV5 iz paketa mdbreact. Naravno ovo se prikazuje ukoliko je ulogovani korisnik admin, a u suprotnom slučaju pisaće ,,You do not have access!“



U komponenti AdminView ukoliko je ulogovan korisnik admin, prikazuje se tabela sa svim buketima, zajedno sa opcijom za edit i brisanje buketa. Ispod same tabele nalazi se dugme za dodavanje novog aranžmana.

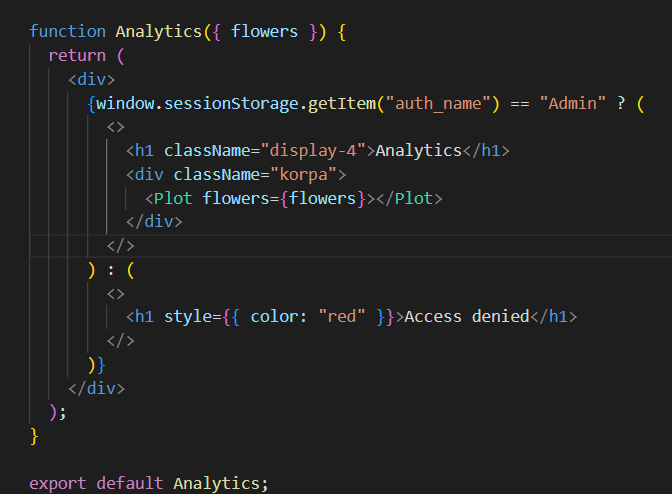




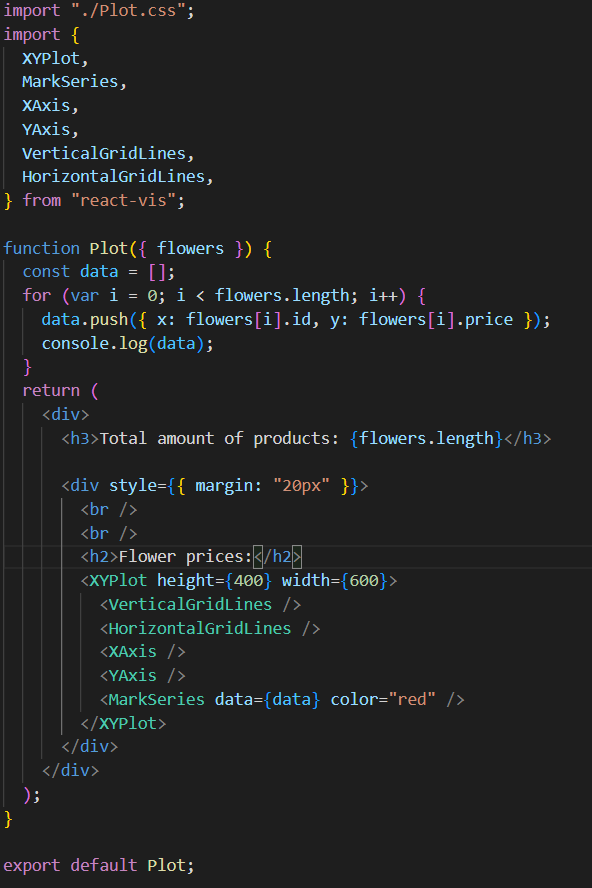
Funkcija deleteFlower je implementirana u App.js uz pomoću axiosa koji se poziva na naš api i sa metodom DELETE.



U komponenti Analytics predstavljeni su pomoću komponente Plot ukupan broj buketa u ponudi a uz to su i grafički predsavljene vrednosti.



Komponenta Plot prikazuje scatter plot cena proizvoda iz ponude korišćenjem komponenata XYPlot, VerticalGridLines, HorizontalGridLines, XAxis, YAxis i MarkSeries iz paketa react-vis. Komponenta MarkSeries prima konstantu data, u kojoj su na x osi id-jevi cveća(buketa), a na y osi njihove cene:



Edit komponenta čini formu koja je uz pomoć axiosa popunjena trenutnim podacima o buketu.



Ova fukcija korišćenjem axios-a zove rutu za izmenu odabranog buketa i u slučaju uspešne izmene prikazuje se poruka o uspešnosti, u suprotnom, u konzoli se ispisuje odgovarajuća greška. Pored ovoga, u ovoj komponenti za učitavanje izabranog buketa i postavljanje dobijene vrednosti na vrednost konstante flowerData, koristi hook useEffect i axios.



Primer svih ruta u App.js:



# Github link ka repozitorijumu

<https://github.com/KacaGrujic/projekatcvecara.git>