Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

WEB prodavnica - sportska oprema

Seminarski rad

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Đurđica | Jovanović | 2019/0446 |
| Jovana | Lazarević | 2019/0417 |
| Branislav | Trajkov | 2019/0229 |
| Link ka GIT-u | https://github.com/djurdjicajov/iteh-projekat.git | |

Sadržaj

[1. Korisnički zahtev 3](#_Toc127879785)

[2. Opis sistema 4](#_Toc127879786)

[2.1. Slučajevi korišćenja 4](#_Toc127879787)

[2.2. Opis arhitekture aplikacije 4](#_Toc127879788)

[2.3. Dijagrami sekvenci 5](#_Toc127879789)

[2.4. Dijagram klasa 7](#_Toc127879790)

[2.5. Specifikacija REST API-ja 7](#_Toc127879791)

[3. Opis tehnologija korišćenih u aplikaciji 14](#_Toc127879792)

[4. Korisničko uputstvo 14](#_Toc127879793)

[5. Prikaz reprezentativnih delova koda 23](#_Toc127879794)

[5.1. Backend 23](#_Toc127879795)

[5.2. Frontend 25](#_Toc127879796)

# Korisnički zahtev

Za realizaciju aplikacije koristili smo React i Laravel, sa elementima i predznanjem iz HTML, CSS, JavaScripta. Prethodno smo instalirali Visual Studio Code i Node.js koji su nam bili potrebni za ovaj rad. U Command Prompt-u smo preko opcije cd prešli u prethodno napravljen folder u kome smo želeli da inicijalizujemo projekat, a nakon toga smo ga inicijalizovali preko opcije npx create-react-app frontend, gde se u folderu frontend inicijalizovala početna struktura aplikacije. U folderu backend kreirani je početni folder za Laravel, gde su kasnije kreirani modeli, kontroleri i rute.

Pokrećemo server preko opcije npm start i na portu 3000 nam se otvara početni ekran naše aplikacije, na kom se nalazi samo react logo koji se pokreće.

Aplikacija je portal Web prodavnica. Neulogovanim korisnicima prikazan je početni ekran gde se nalazi slide show slika sportske opreme i na dnu stranice footer o osnovnim informacijama o kompaniji, lokacijama i linkovima ka Početnoj stranici, Ponudi, Korpi i Kontakt stranici. Stranica Ponuda i Korpa su omogućene samo za ulogovane korisnike.

U navigacionom meniju nalaze se stranice Uloguj se i Registruj se u zavisnosti od toga da li korisnik već ima nalog. Ukoliko korisnik nema napravljen nalog, nakon registrovanja biće prebačen na stranicu Uloguj se. Ulogovani korisnik se automatski prebacuje na stranicu Ponuda, gde može videti sve proizvode koje imamo u asortimanu i može ih sortirati rastuće ili opadajuće po ceni. Korisnik može proizvode dodavati i unbacivati u korpu, a klikom na stranicu Korpa prikazaće mu se svi proizvodi koje je dodao u Korpu, gde ima opciju da poveća ili smanji količinu, kao i da izbaci neke od proizvoda iz korpe. Takođe, može promeniti valuta iz dinara u evre ili dolare i generisati račun u izabranoj valuti. Klikom na Generiši račun, generiše se račun u PDF formatu. Na stranici Kontakt korisnik može da ostavi komentar, odnosno sugestiju ili utisak. Klikom na Odjavi se, korisnik je izlogovan i vraća se na Početnu stranicu.

Ukoliko se korisnik uloguje kao admin, imaće pristup izmeni proizvoda, odnosno brisanju i izmeni postojećih proizvoda.

Takođe imaće uvid u sve poruke poslate preko Kontakt stranice i uvid u Analitiku, odnosno statistiku kupovine proizvoda. Kao I ostali ulogovani korisnici I admin ima mogućnost da se odjavi I vrati na Početnu stranicu.

# Opis sistema

## 2.1. Slučajevi korišćenja

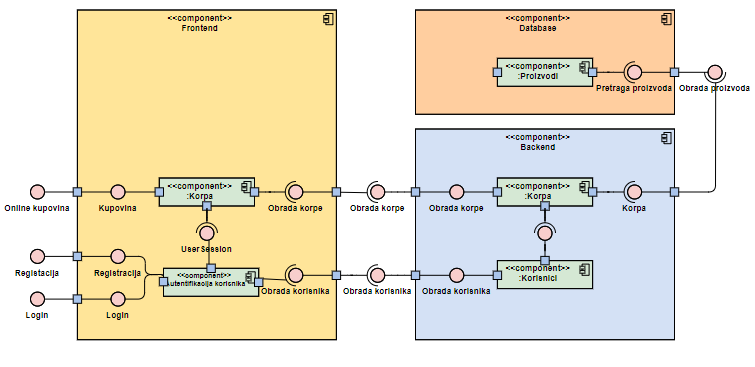
Slučajevi korišćenja:

1. Registracija
2. Login
3. Logout
4. Prikaz Ponude
5. Izmena Ponude
6. Brisanje Ponude
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 – statitiska proizvoda

## 2.2. Opis arhitekture aplikacije

Aplikacija se sastoji od tri celine: frontend, backend i baze podataka koje su povezane i komuniciraju pomoću svojih komponenti. Backend dobija podatke iz baze pomoću MySQL sistema za upravljanje bazama podataka. Podaci potrebi za povezivanje sa bazom se nalaze u .env fajlu u okvuru backanda-a. Komunikacija frontenda-a sa backend-om omogućena je korišćenjem Axios.

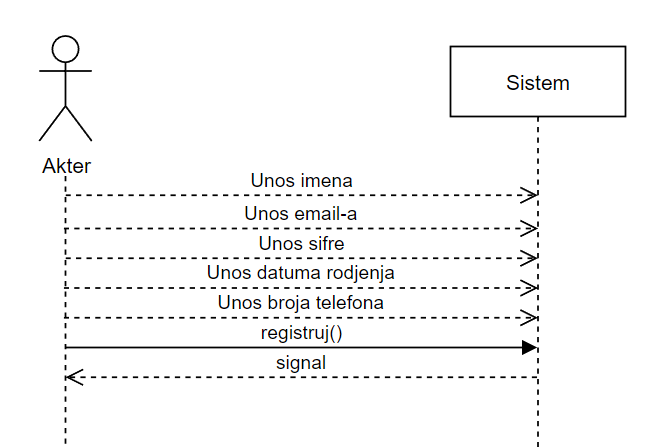
Za razvoj apikacije korišćen je MVC (Model-View-Controller) patern, kojeg karatkeriše postojanje kontrolera koji reaguje na akcije korisnika.



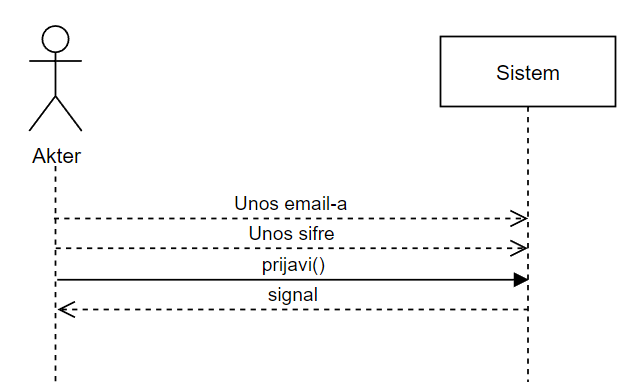
## 2.3. Dijagrami sekvenci

Za neke od već opisanih slučajeva korišćenja aplikacije u nastavku su dati odgovarajući dijagrami sekvenci:

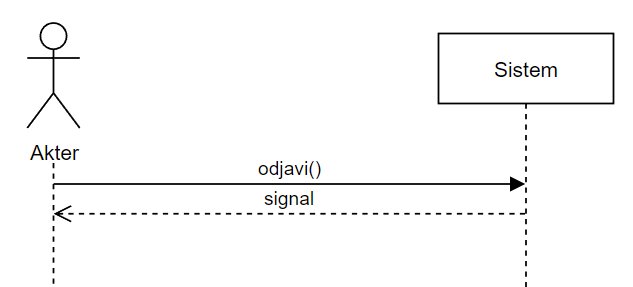
1. Registracija



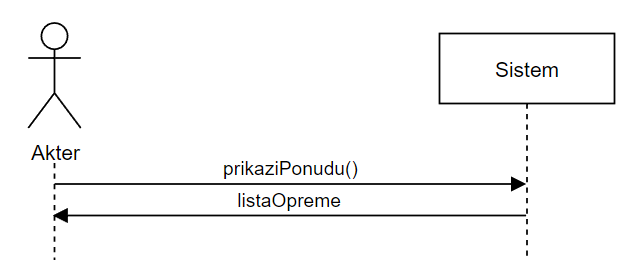
1. Login



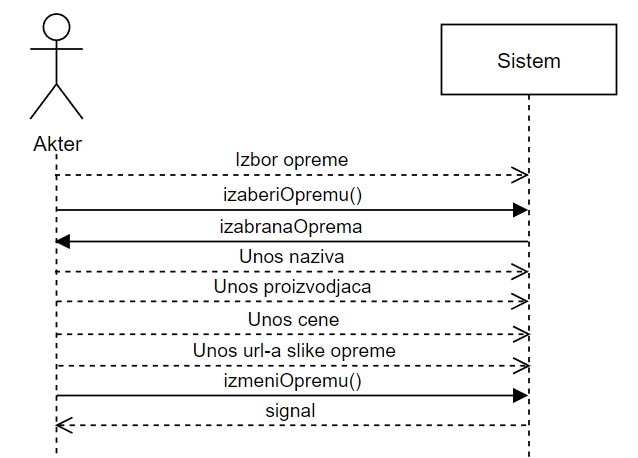
1. Logout



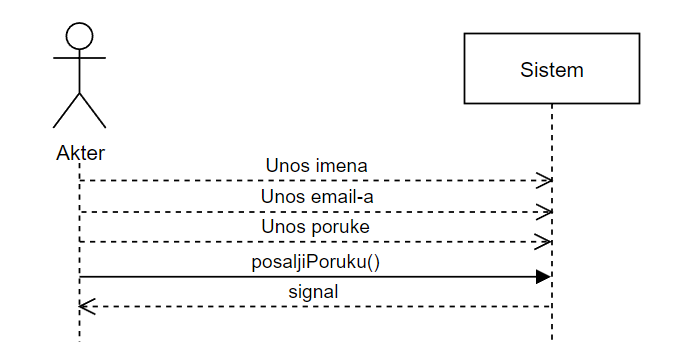
1. Prikaz Ponude



1. Izmena Ponude



11. Slanje poruke



## 2.4. Dijagram klasa

Diagram

Description automatically generated

## Specifikacija REST API-ja

|  |  |
| --- | --- |
| Opisfunkcije | 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" : "Djurdjica ",      "email": "djurdjica@dj.com",      "password" : "1234"      "phone" : "0611234567"      "birthdate" : "13.06.2000."  } |
| Izlazniparametri | {      "data": {          "name": "Djurdjica",          "email": "djurdjica@dj.com",          "updated\_at": "2023-02-20T13:04:26.000000Z",          "created\_at": "2023-02-20T13:04:26.000000Z",          "id": 12      },      "acess\_token": "1|evOQQDA6zUbmZgns6oYse993gbeQNBv9FaSJdFh8",      "token\_type": "Bearer"  } |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Logovanje korisnika na sistem |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | Nema |
| HTTP body parametri | email, password |
| Format HTTP body parametara | {      "email" : "djurdjica@dj.com",      "password" : "1234"  } |
| Izlazniparametri | {      "0": {          "status": 200,          "username": "djurdjica",          "token": "2|8aHktyo9j08CHVaBlRxg622ENvZovuJeTUA0LeR8",          "role": " ",          "id": 12      },      "success": **true**  } |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Odjavljivanjekorisnika sa Sistema |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | [  [      "Uspešno ste se izlogovali!"  ] |
| Format izlaznihparametara | application/json |
| Opis funkcije | Prikaz poruka |
| HTTP metoda | GET |
| URL | /api/poruke |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | [      {          "id": 1,          "ime": "Djurdjica",          "email": "djurdjica@dj.com",          "poruka": "Da li imate patike u broju 37?",          "created\_at": "2023-02-20T13:14:50.000000Z",          "updated\_at": "2023-02-20T13:14:50.000000Z"      }  ] |
| Format izlaznihparametara | application/json |
|  |  |
| Opisfunkcije | Prikaz proizvoda |
| HTTP metoda | GET |
| URL | /api/proizvod |
| URL parametri | nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | {      "data": [          {              "id": 1,              "naziv": "Nike Downshifter 12",              "proizvodjac": "Nike",              "zemlja\_porekla": "Srbija",              "cena": 6500,              "kolicina": 0,              "kategorija": {                  "id": 1,                  "naziv": "Patike"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/DD9/DD9294-001/images/thumbs\_w/thumbs\_800/DD9294-001\_w\_800\_800px.jpg"          },          {              "id": 2,              "naziv": "Nike Get Fit",              "proizvodjac": "Nike",              "zemlja\_porekla": "Srbija",              "cena": 6300,              "kolicina": 0,              "kategorija": {                  "id": 2,                  "naziv": "Dukserica"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/DQ5/DQ5558-503/images/thumbs\_w/thumbs\_800/DQ5558-503\_2\_w\_800\_800px.jpg"          },          {              "id": 3,              "naziv": "Nike Sportswear Therma-FIT Windrunner",              "proizvodjac": "Nike",              "zemlja\_porekla": "Srbija",              "cena": 16000,              "kolicina": 0,              "kategorija": {                  "id": 3,                  "naziv": "Jakna"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/DH4/DH4073-010/images/thumbs\_w/thumbs\_800/DH4073-010\_w\_800\_800px.jpg"          },          {              "id": 4,              "naziv": "adidas TANGO ROSARIO",              "proizvodjac": "adidas",              "zemlja\_porekla": "Srbija",              "cena": 2700,              "kolicina": 0,              "kategorija": {                  "id": 4,                  "naziv": "Lopta"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/656/656927/images/thumbs\_w/thumbs\_800/656927\_w\_800\_800px.jpg"          },          {              "id": 5,              "naziv": "THE NORTH FACE COORDINATES",              "proizvodjac": "THE NORTH FACE",              "zemlja\_porekla": "Srbija",              "cena": 4990,              "kolicina": 0,              "kategorija": {                  "id": 5,                  "naziv": "Majica"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/NF0/NF0A7X2KJK31/images/thumbs\_w/thumbs\_800/NF0A7X2KJK31\_w\_800\_800px.jpg"          },          {              "id": 6,              "naziv": "adidas Basic 3-Stripes",              "proizvodjac": "adidas",              "zemlja\_porekla": "Srbija",              "cena": 8000,              "kolicina": 0,              "kategorija": {                  "id": 6,                  "naziv": "Trenerka"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/IC6/IC6755/images/thumbs\_w/thumbs\_800/IC6755\_w\_800\_800px.jpg"          },          {              "id": 7,              "naziv": "PUMA VIKKY STACKED",              "proizvodjac": "Puma",              "zemlja\_porekla": "Srbija",              "cena": 8000,              "kolicina": 0,              "kategorija": {                  "id": 1,                  "naziv": "Patike"              },              "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/369/369144-01/images/thumbs\_w/thumbs\_800/369144-01\_w\_800\_800px.jpg"          }      ]  } |
| Format izlaznihparametara | application/json |
|  |  |
| Opisfunkcije | Dodavanje novog proizvoda |
| HTTP metoda | POST |
| URL | /api/proizvod |
| URL parametri | Nema |
| HTTP body parametri | Naziv,proizvodjac,cena,kategorija,image,zemlja\_porekla |
| Format HTTP body parametara | {      "naziv": "Model123",      "proizvodjac": "Nike",      "cena": "4999"      "kategorija": "2",      "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/137/1377428-001/images/thumbs\_w/thumbs\_800/1377428-001\_w\_800\_800px.jpg",      "zemlja\_porekla": "Kina"  } |
| Izlazniparametri | [      "Uspesno kreiran proizvod",      {          "naziv": "Model123",          "proizvodjac": "Nike",          "cena": "4999",          "zemlja\_porekla": "Kina",          "kolicina": 0,          "kategorija": "2",          "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/137/1377428-001/images/thumbs\_w/thumbs\_800/1377428-001\_w\_800\_800px.jpg",          "updated\_at": "2023-02-20T13:37:23.000000Z",          "created\_at": "2023-02-20T13:37:23.000000Z",          "id": 8      }  ] |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Prikazivanje jednog proizvoda |
| HTTP metoda | GET |
| URL | /api/proizvod/{proizvodId} |
| URL parametri | proizvodId |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | {      "data": {          "id": 3,          "naziv": "Nike Sportswear Therma-FIT Windrunner",          "proizvodjac": "Nike",          "zemlja\_porekla": "Srbija",          "cena": 16000,          "kolicina": 0,          "kategorija": {              "id": 3,              "naziv": "Jakna"          },          "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/DH4/DH4073-010/images/thumbs\_w/thumbs\_800/DH4073-010\_w\_800\_800px.jpg"      }  } |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Prikazivanje stavki korpi |
| HTTP metoda | GET |
| URL | /api/stavke |
| URL parametri | nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | [      {          "id": 1,          "korpa\_id": 1,          "proizvod\_id": 1,          "kolicina": 2,          "created\_at": "2023-02-19T14:43:27.000000Z",          "updated\_at": "2023-02-21T14:43:27.000000Z"      }  ] |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Brisanje odabranog proizvoda |
| HTTP metoda | DELETE |
| URL | /api/proizvod/{proizvodId} |
| URL parametri | ProizvodId |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | "uspesno obrisano!" |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Ažuriranje proizvoda |
| HTTP metoda | PUT/PATCH |
| URL | /api/proizvod/{proizvodID} |
| URL parametri | proizvodID |
| HTTP body parametri | Naziv,proizvodjac,cena,kategorija,image,zemlja\_porekla |
| Format HTTP body parametara | {      "naziv": "Model123",      "proizvodjac": "Nike",      "cena": "4999"      "kategorija": "2",      "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/137/1377428-001/images/thumbs\_w/thumbs\_800/1377428-001\_w\_800\_800px.jpg",      "zemlja\_porekla": "Kina"  } |
| Izlazniparametri | {      "success": **true**,      "message": "Proizvod uspesno izmenjen.",      "data": {          "id": 2,          "naziv": "Model123",      "proizvodjac": "Nike",      "cena": "4999"      "kategorija": "2",      "image": "https://www.sportvision.rs/files/thumbs/files/watermark/files/images/slike\_proizvoda/media/137/1377428-001/images/thumbs\_w/thumbs\_800/1377428-001\_w\_800\_800px.jpg",      "zemlja\_porekla": "Kina"      }  "kolicina": "0"    } |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Slanje poruka |
| HTTP metoda | POST |
| URL | /api/kontakt |
| URL parametri | Nema |
| HTTP body parametri | name |
| Format HTTP body parametara | {              "ime": "Djurdjica",  "poruka": "Imete li Nike patike 37?",  "email": "djurdjica@dj.com",                      } |
| Izlazniparametri | {      "success": **true**} |
| Format izlaznihparametara | application/json |
|  |  |
| Opisfunkcije | Prikazivanje profila |
| HTTP metoda | GET |
| URL | /api/profile |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | {      "id": 12,      "name": "bike",      "email": "djurdjica@dj.com",      "email\_verified\_at": **null**,      "created\_at": "2023-02-20T13:04:26.000000Z",      "updated\_at": "2023-02-20T13:04:26.000000Z",      "phone": "+38161 5424 9544",      "birthdate": "1987-04-06",      "admin": 0  } |
| Format izlaznihparametara | application/json |
| Opisfunkcije | Admin ulogovan |
| HTTP metoda | GET |
| URL | /api/check |
| URL parametri | Nema |
| HTTP body parametri | Nema |
| Format HTTP body parametara | Nema |
| Izlazniparametri | {      "message": "Admin ulogovan"  } |
| Format izlaznihparametara | application/json |

# Opis tehnologija korišćenih u aplikaciji

Laravel je open-source web framework. Baziran je na PHP-u, odnosno Simfony framework-u. Razvio ga je Taylor Otwell s ciljem razvoja web aplikacija prateći model–view–controller (MVC) arhitekturu. Smatra se jednim od najpopularnijih PHP framework-a. On olakšava razvoj razvoj delova aplikacije kao što su autentifikacija, rutiranje, sesije i keširanje.

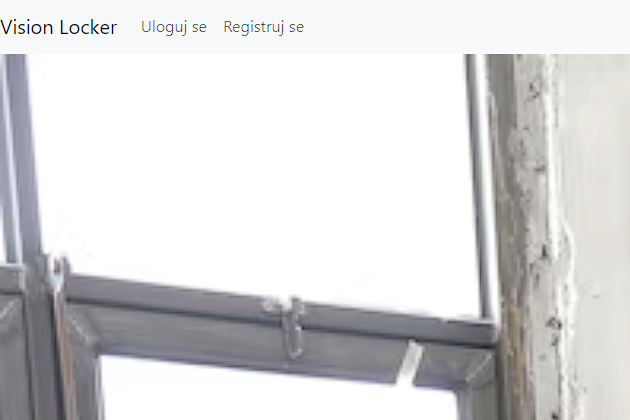
Axios je biblioteka koja služi za slanje zahteva API-u, vraća podatke I obrađuje ih tj. ona komunicira sa backend delom aplikacije.

React, (poznat i kao React.js ili ReactJS ) je JavaScript biblioteka kreirana sa ciljem izgradnje korisničkih interfejsa. Stvorio ga je Jordan Valke, programer u kompaniji Fejsbuk. Od svog debija 2011. godine, React je postao jedan od najpopularnijih front-end framework-a za izradu veb aplikacija sa ogromnom zajednicom pojedinačnih programera i kompanija.

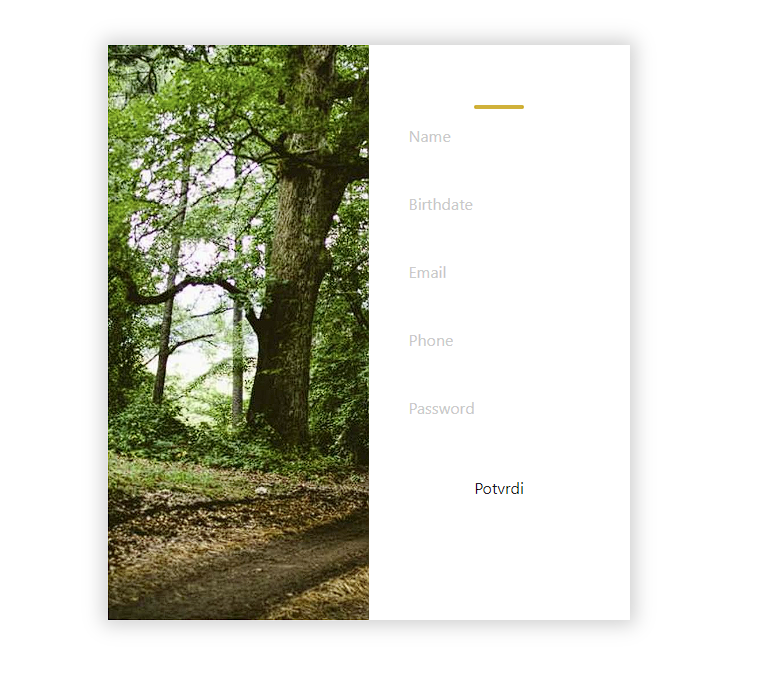
# Korisničko uputstvo

U skladu sa navedenim slučajevima korišćenja, u uputstvu za svaki od njih objašnjeni su koraci kako bi korisnik lako mogao pristupiti korišćenju aplikacije:

1. Registracija



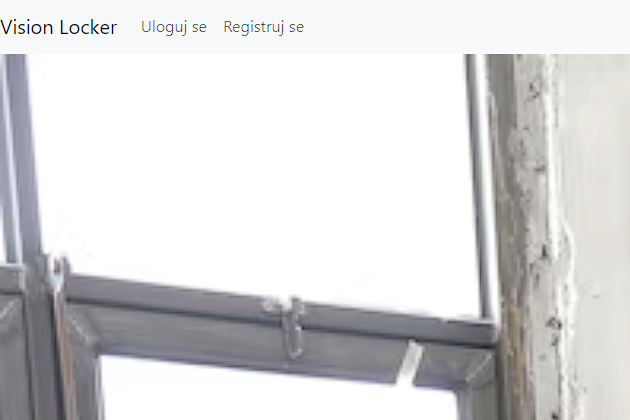
Na početnoj stranici prvo je potrebno kliknuti na dugme „Registruj se“ u gorenjem levom uglu, kako bi se otvorila stranica za registraciju.

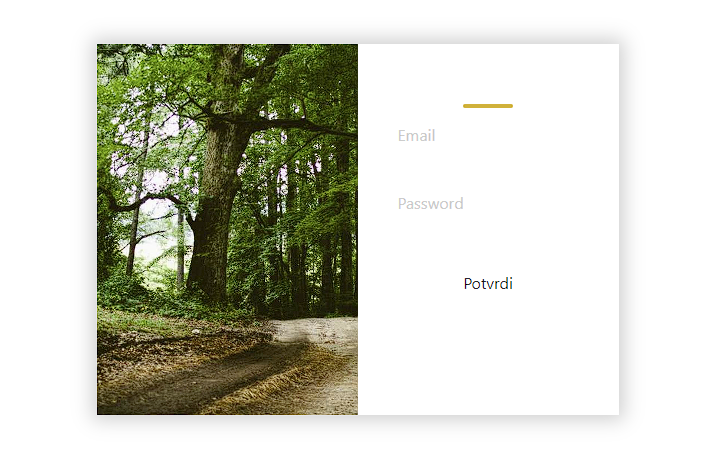


U formi koja se prikazuje, korisnik treba da unese sve potrebne podatke za registraciju, nakon čega klikće dugme „Potvrdi“, nakon čega se odmah prebacuje na stranicu za prijavu.

1. Login

Kada korisnik želi da se uloguje u aplikaciji potrebno je da u gornjem levom uglu klikne na dugme „Uloguj se“, čime se otvara stranica za logovanje.

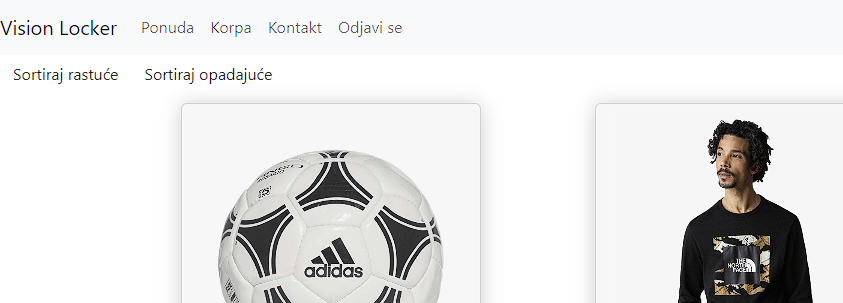




Na ovoj stranici u formi je potrebno uneti podatke za prijavu, odnosno email i šifru, a nakon toga kliknuti na dugme „Potvrdi“, nakon čega je korisnik prijavljen.

1. Logout

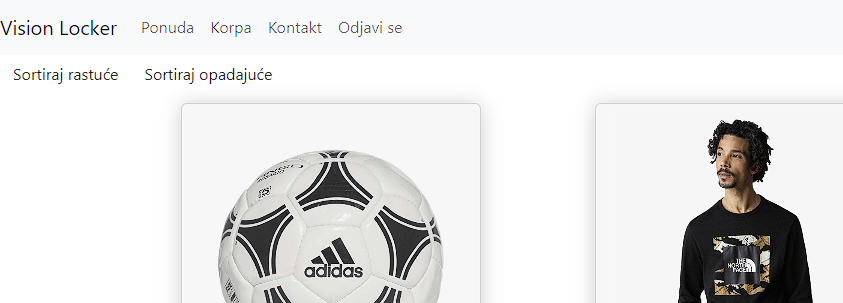
Kada prijavljeni korisnik želi da se odjavi iz aplikacije, potrebno je samo da klikne na dugme „Odjavi se“, koje se nalazi u gornjem levom uglu, na kojoj kod stranici da se korisnik nalazi.



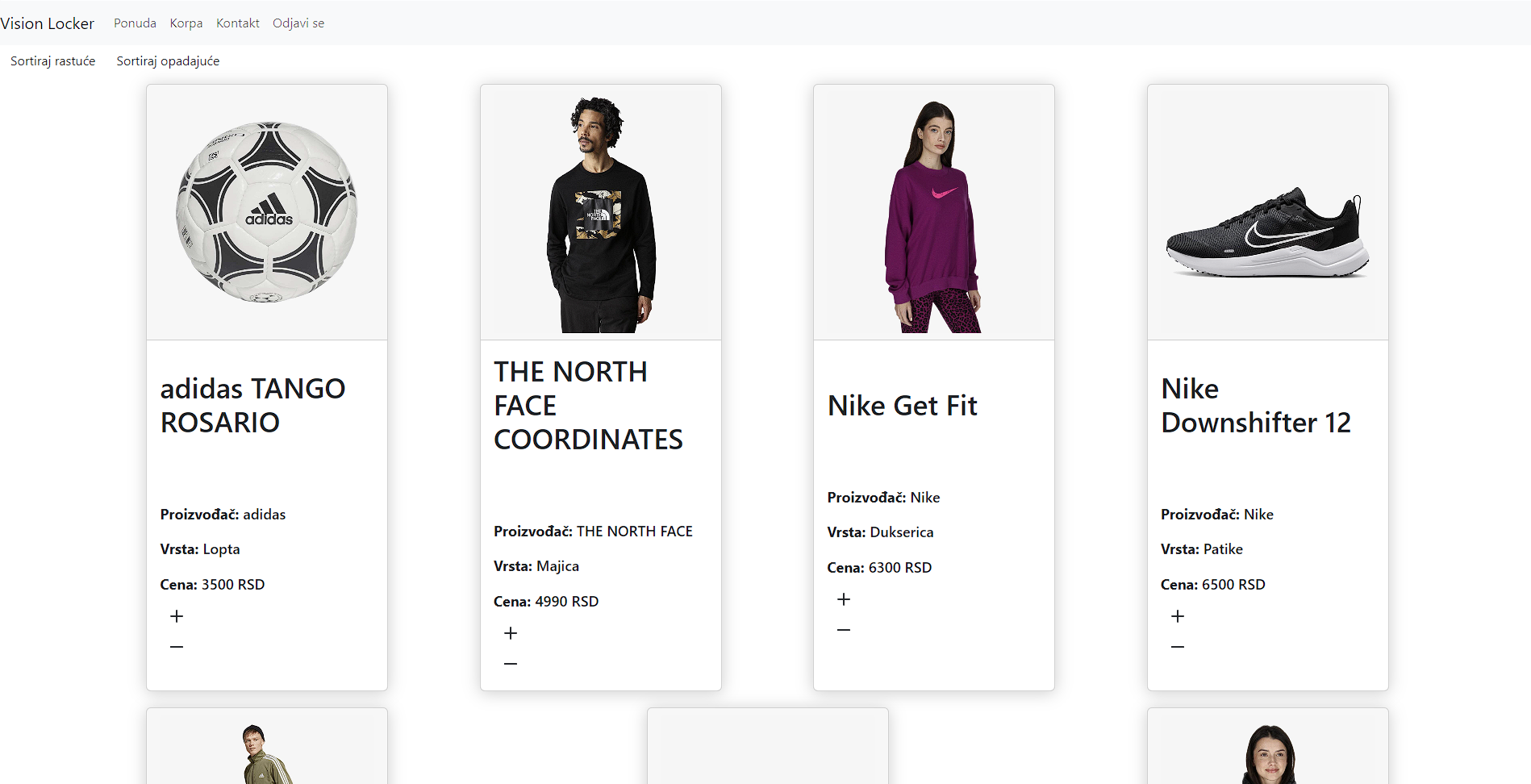
Nakon toga, korisnik je odjavljen iz aplikacije i prikazuje mu se početna stranica.

1. Prikaz Ponude

Kako bi se korisniku prikazala ponuda, odnosno lista svih proizvoda, potrebno je da klikne dugme „Ponuda“ koje se nalazi u gornjem levom uglu stranice, gde god da se nalazi na sajtu:

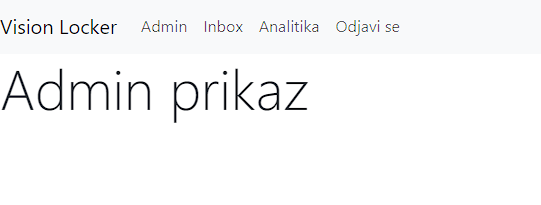


Nakon ovoga, korisniku se prikazuje stranica sa listom svih proizvoda:

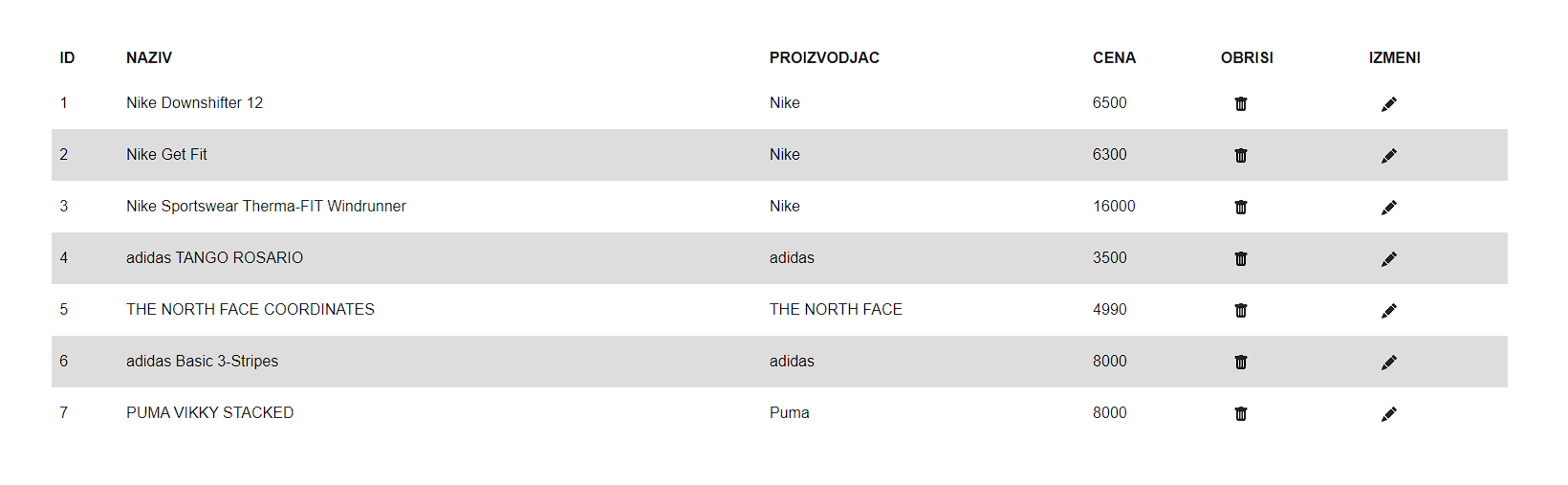


1. Izmena Ponude

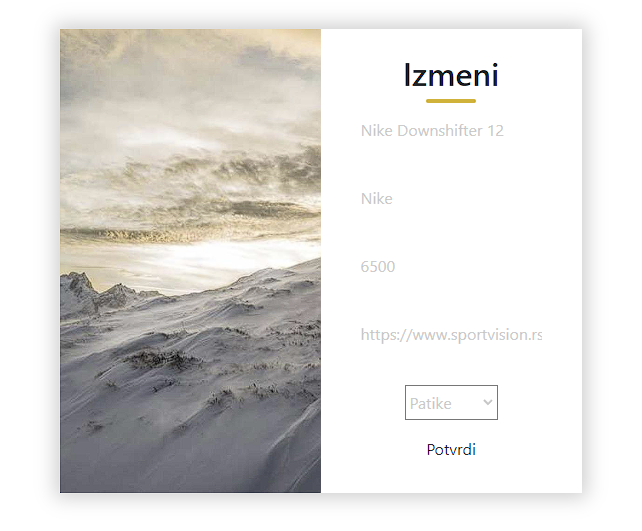
Kako bi menjao neki proizvod iz ponude, korisnik mora biti ulogovan kao admin. Nakon toga, potrebno je kliknuti na dugme „Admin“ u gornjem levom uglu stranice:



Time se otvara spisak svih proizvoda zajedno sa dugmetom za izmenu:



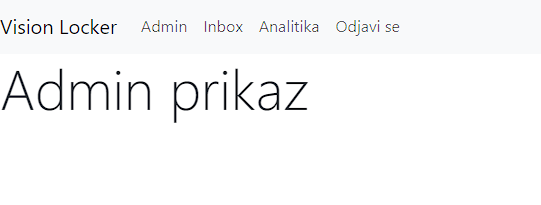
Korisnik zatim treba da klikne na dugme za izmenu onog proizvoda koga želi da izmeni čime se otvara stranica za izmenu proizvoda:



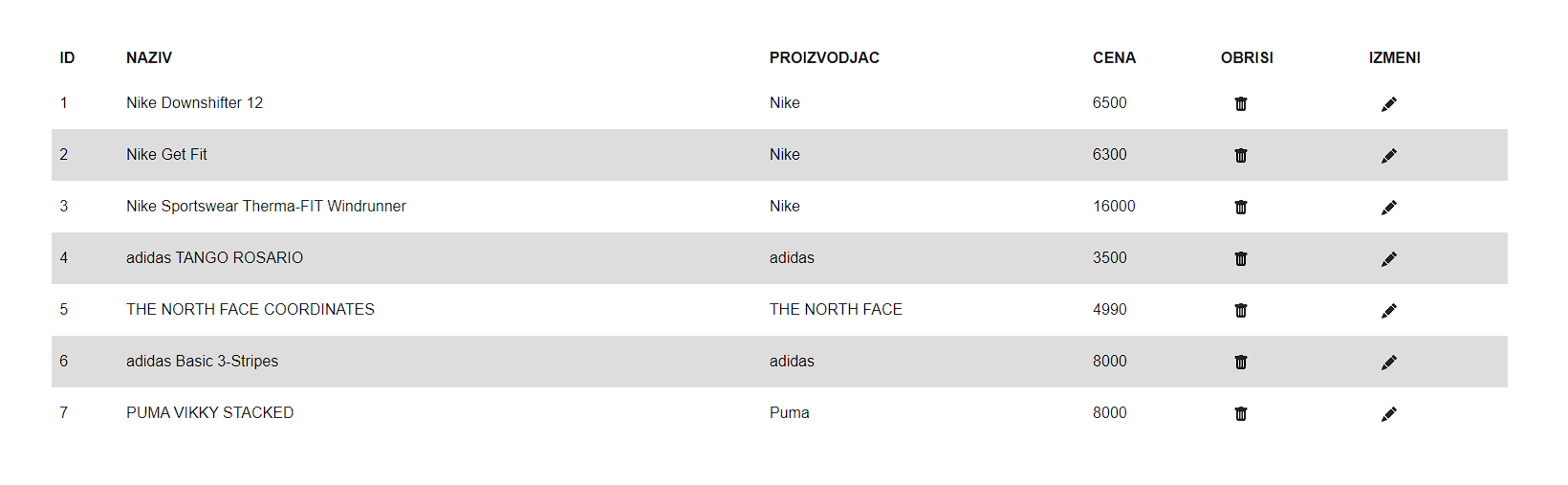
Korisnik zatim unos i menja podatke o proizvodu i nakon toga klikom na dugme „Potvrdi“ čuva unete podatke o proizvodu u aplikaciji.

1. Brisanje Ponude

Kako bi obrisao neki proizvod iz ponude, korisnik mora biti ulogovan kao admin. Nakon toga, potrebno je kliknuti na dugme „Admin“ u gornjem levom uglu stranice:



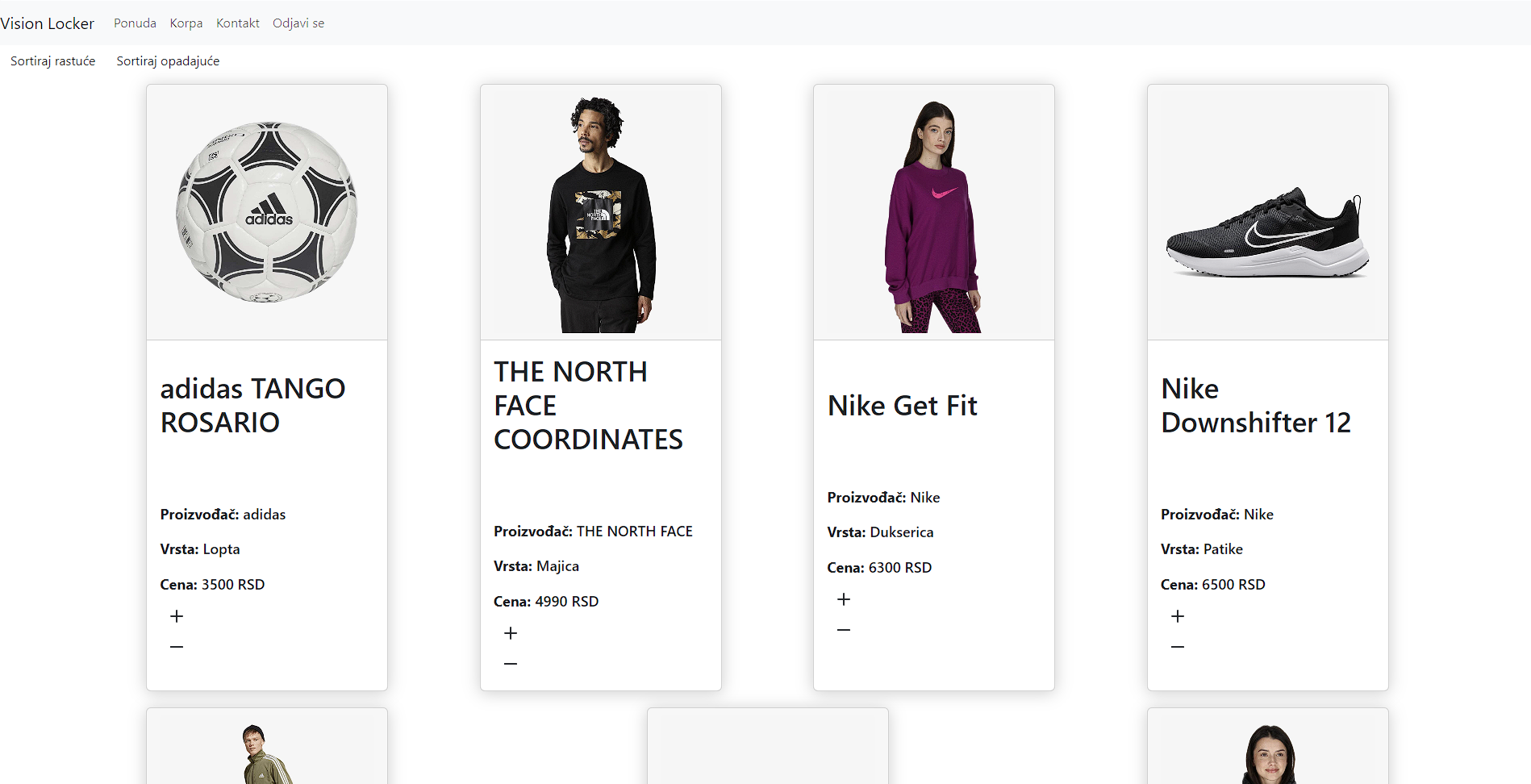
Time se otvara spisak svih proizvoda zajedno sa dugmetom za brisanje:



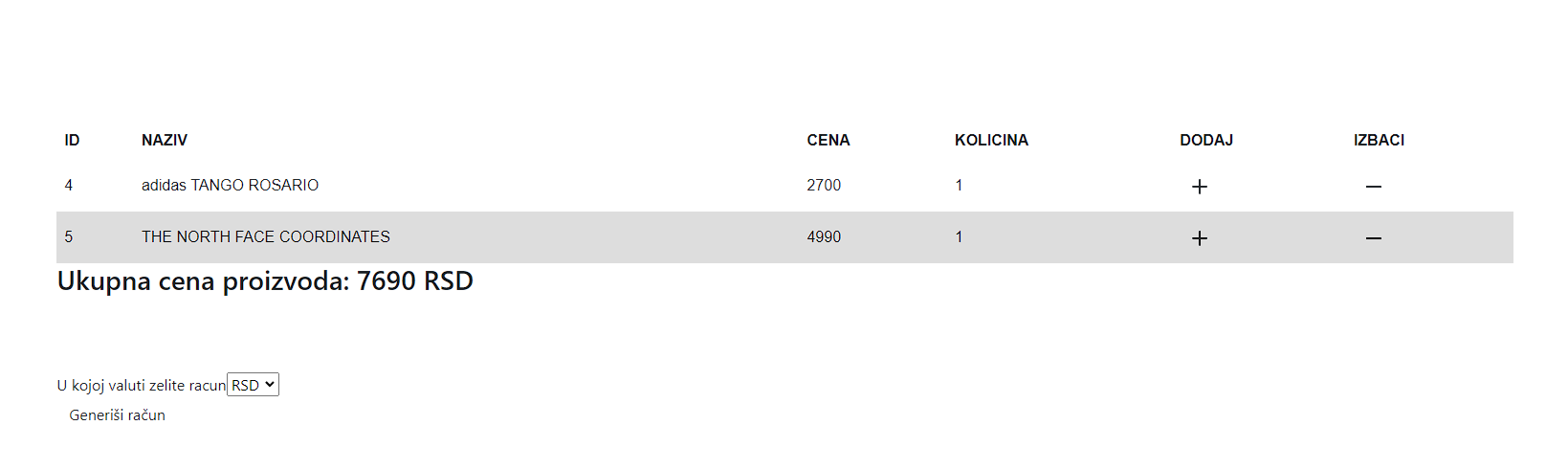
Korisnik bira proizvod koji želi da obriše i zatim klikće na dugme za brisanje tog proizvoda, čime se briše proizvod iz ponude.

1. Dodavanje proizvoda u korpu

Korisnik dodaje proizvode u korpu klikom da dugme „+“ u okviru kartice određenog proizvoda iz korpe:

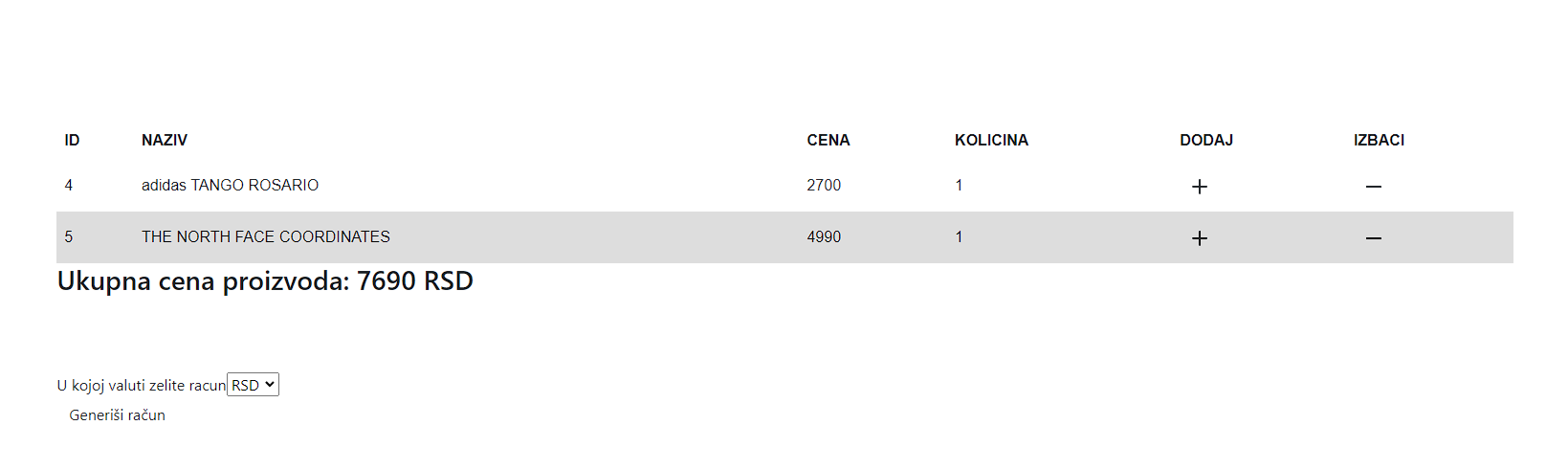


Sve dodate proizvode u korpi, korisnik može videti klikom na dugme „Korpa“ u gornjem levom uglu, čime se otvara stranica sa svim proizvodima u korpi:



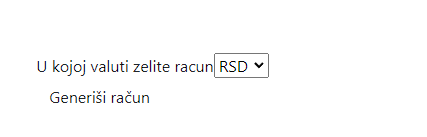
1. Brisanje proizvoda iz korpe

Kada korisnik želi da obriše proizvod/e iz korpe potrebno je samo da klikne na dugme „-“ u koloni izbaci, kako bi smanjio količinu konkretnog proizvoda, odnosno izbacio ga iz korpe, ako se samo jedan takav proizvod nalazi u korpi:



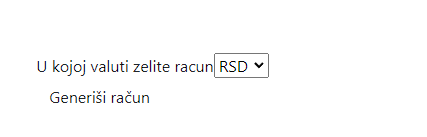
1. Izbor valute

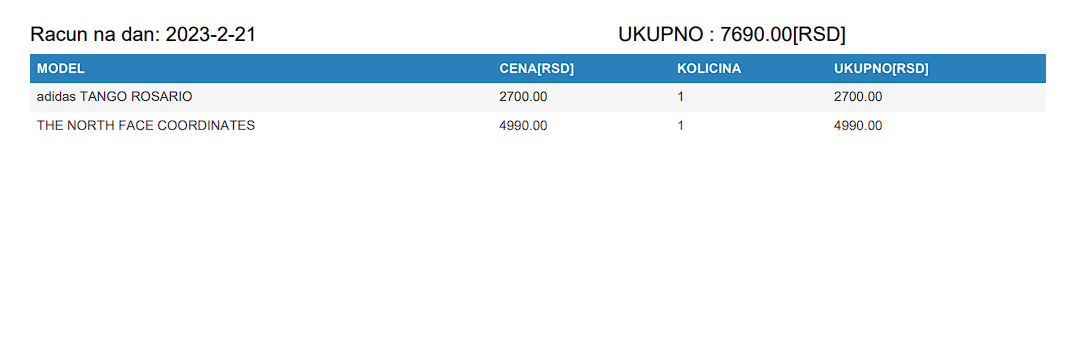
Kada je korisnik završio sa odabirom proizvoda, kako bi odabrao valutu potrebno je da iz padajuće liste izabere željenu valutu kao na slici:



1. Generisanje računa

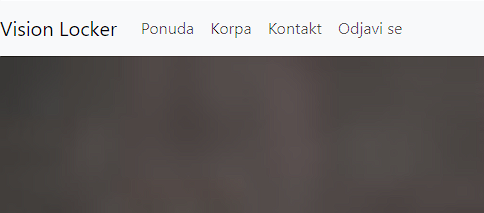
Kada je korisnik dodao željene proizvode u korpu i izabrao valutu u kojoj želi da mu bude ispostavljen račun, klikom na dugme „Generiši račun“ korisniku se preuzima račun u pdf formatu:



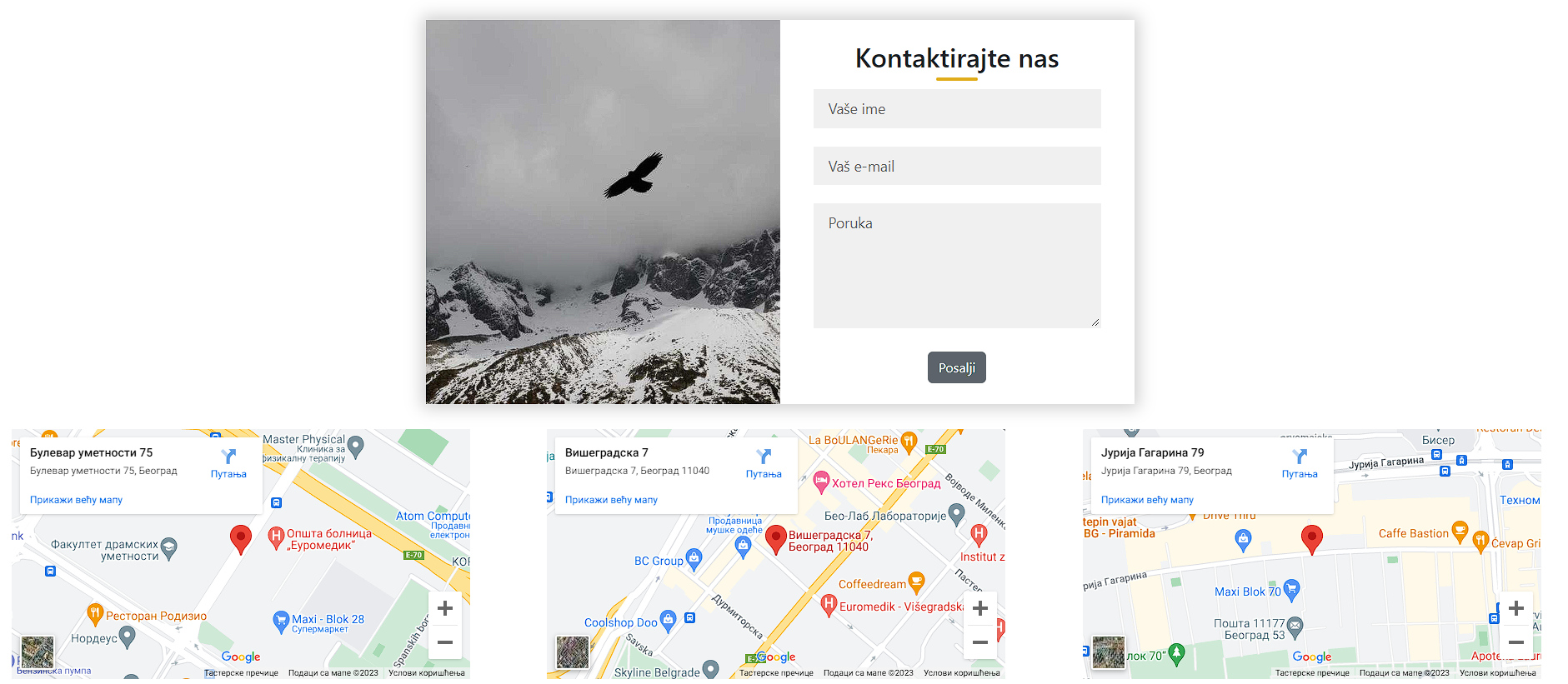


1. Slanje poruke

Korisnik prvo mora kliknuti na dugme „Kontakt“ u gornjem levom uglu:



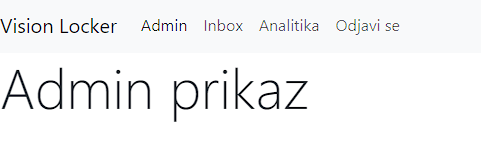
Nakon toga, otvara mu se kontakt stranica sa formom za slanje poruke:



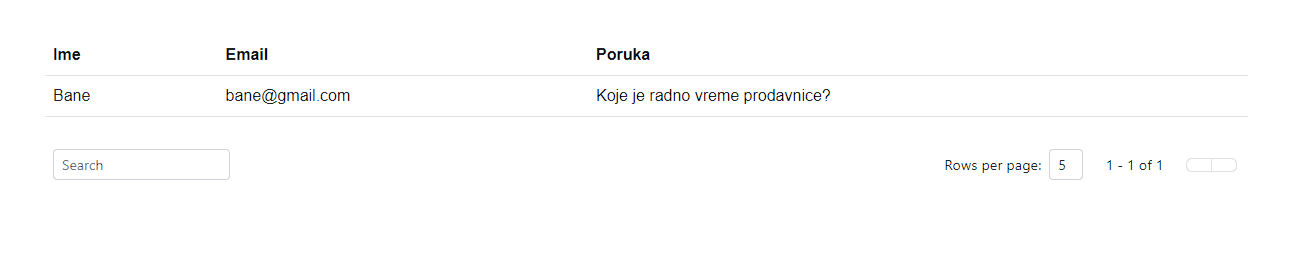
Korisnik unosi ime, email i poruku koju želi poslati prodavnici i kako mi poslao poruku, potrebno je da klikne na dugme „Pošalji“, čime se poruka šalje prodavnici.

1. Pregled poruka

Pravo na pristup listi primljenih poruka prodavnice ima samo admin. Nakon logovanja, potrebno je kliknuti dugme „Inbox“ u gornjem levom uglu:

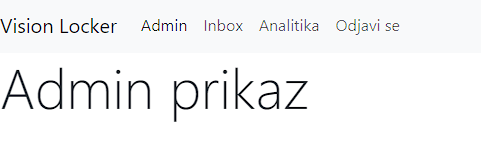


Nakon toga se otvara inbox stranica sa listom svih poruka i informacija o njoj:



1. Pregled Analitike – statistika proizvoda

Kao i u slučaju pregleda poruka, pravo pregleda analitike ima samo admin. U gornjem levom uglu potrebno je kliknuti dugme „Analitika“, čime se otvara stranica analitike:





# Prikaz reprezentativnih delova koda

## Backend

Pomoću laravela impementirali smo sledeće modele: User, Proizvod, Kategorija, Korpa, StavkaKorpe, Poruka.

User:

class User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array<int, string>

     \*/

    protected $fillable = [

        'name',

        'email',

        'password',

    ];

    /\*\*

     \* The attributes that should be hidden for serialization.

     \*

     \* @var array<int, string>

     \*/

    protected $hidden = [

        'password',

        'remember\_token',

    ];

    /\*\*

     \* The attributes that should be cast.

     \*

     \* @var array<string, string>

     \*/

    protected $casts = [

        'email\_verified\_at' => 'datetime',

    ];

}

Proizvod:

class Proizvod extends Model

{

    use HasFactory;

    protected $fillable = [

        'image',

        'naziv',

        'proizvodjac',

        'cena',

        'kolicina',

        'kategorija',

        'zemlja\_porekla'

      ];

    public function kategorija()

    {

       return $this->belongsTo(Kategorija::class);

    }

    public function stavke()

    {

       return $this->belongsTo(StavkaKorpe::class);

    }

}

Korpa:

class Korpa extends Model

{

    use HasFactory;

    protected $fillable = [

        'user\_id',

    ];

    public function stavke()

    {

        return $this->hasMany(StavkaKorpe::class);

    }

}

Stavka korpe:

class StavkaKorpe extends Model

{

    use HasFactory;

    protected $fillable = [

        'proizvod\_id',

        'korpa\_id',

        'kolicina',

    ];

    public function korpa()

    {

        return $this->belongsTo(Korpa::class);

    }

    public function proizvod(){

        return $this->belongsTo(Proizvod::class);

    }

}

Kategorija:

lass Kategorija extends Model

{

    use HasFactory;

    protected $fillable = [

        'naziv'

    ];

    public function proizvodi(){

        return $this->hasMany(Proizvod::class);

    }

}

Poruka:

class Poruka extends Model

{

    use HasFactory;

    protected $fillable = [

        'ime',

        'email',

        'poruka'

      ];

}

U aplikaciji su koršćene sledeće API rute:

Route::post('/register', [AuthController::class, 'register']);

Route::post('/login', [AuthController::class, 'login']);

Route::post('/kontakt', [PorukaController::class, 'primiPoruku']);

Route::get('/poruke', [PorukaController::class, 'index']);

Route::get('/proizvod', [ProizvodController::class, 'index']);

Route::post('/proizvod', [ProizvodController::class, 'store']);

Route::put('/proizvod/{id}', [ProizvodController::class, 'update']);

Route::delete('/proizvod/{id}', [ProizvodController::class, 'destroy']);

Route::get('/proizvod/{id}', [ProizvodController::class, 'show']);

Route::resource('stavke', StavkaKorpeController::class);

Route::group(['middleware' => ['auth:sanctum']], function () {

    Route::get(

        '/profile',

        function (Request $request) {

            return auth()->user();

        }

    );

    Route::resource('korpe', KorpaController::class);

    Route::post('/logout', [AuthController::class, 'logout']);

});

Route::middleware(['auth:sanctum', 'isAdmin'])->group(function () {

    Route::get(

        '/check',

        function () {

            return response()->json(['message' => 'Admin ulogovan'], 200);

        }

    );

## Frontend

Na početku smo, na način koji je gore naveden, inicijalizovali početnu strukturu projekta, koja se sastoji iz node-modules foldera (koji kroz rad nismo direktno koristili i menjali), public foldera (u kom su po default-u smeštena 2 react logoa različite veličine-koje takođe nismo korisitli ali ih nismo ni obrisali, kao i index.html stranica koju smo na početku malo izmenili), src folder(u kom su smeštene sve komponente koje smo koristili kroz rad), kao i dva .json fajla koje isto direktno nismo dirali, već ukoliko bi došlo do neke instalacije, došlo bi i do promene neke linije koda ili dodavanje nove linije u ta dva .json fajla.

Prvo što smo uradili jeste kreiranje NavBar-a i započeto kreiranje Početne stranice koju smo kasnije dovršili.

Text

Description automatically generated

Text

Description automatically generated

Takođe dodali smo izmene i na stranici App.js gde smo importovali NavBar i dodali rutu NavBar.

Zatim smo doradili Početnu stranicu na kojoj smo dodali prikaz slika u vidu slajdova, za koji je bilo potrebno da instaliramo react-slideshow-image i importujemo na stranicu Pocetna.

A picture containing text

Description automatically generated

Zatim smo dodali stranicu Footer koja se ponavlja na svakoj stranici i sadrži osnovne informacije o kompaniji, lokacije i brze linkove ka određenim stranicama.

Text

Description automatically generated

U App.js -u smo importovali Footer i Početnu i dodali rute za njih.

Takođe smo u App.css-u kreirali početni dizajn koji smo kasnije samo dorađivali.

Graphical user interface, text

Description automatically generated

Zatim smo napravili stranice Login I Register i uredili njihov dizajn. Kako bismo povezali React i Laravel bilo je potrebno da instaliramo axios. Pored axiosa instalirali smo i react-router-dom.

Na stranici za Login improtovali smo axcios i react-router-dom.

Text

Description automatically generated

Povezali smo stranicu sa Laravel rutom za login, kako bi prepoznao registrovane korisnike iz baze i omogućio im pristup. Takođe proverava se da li je korisnik admin i ukoliko jeste prebacuje na prikaz za admina, u suprtonom na stranicu Ponude. Ukoliko korisnik nije registrovan, izlazi poruka neuspesno.

Text

Description automatically generated

Po sličnom principu kreirana je i stranica Register.

Text

Description automatically generated

Povezana je sa rutom ka registrovanju kako bi se sačuvali korisnici u bazi, nakon čega prebacuje korisnike na stranicu Login.

Forma za registrovanje zahteva podatke o imenu,rođendanu,email-u, broju telefona i lozinci. Dok forma za login zahteva email i lozinku. Text

Description automatically generated

Dizajn ovih stranica kreiran je u LoginRegister.css fajlu gde smo omogućili smenjivanje nasumičnih slika na formi.

Text

Description automatically generated

Zatim smo napravili fajlove Oprema.jsx i OpremaKartica.jsx uz pomoć kojih je omogućen pristup proizvodima za ulogovane korisnike za podacim o nazivu,ceni, proizvođaču i kategoriji proizvoda. Prilikom kreiranja ovih fajlova instalirali smo i importovali react-icons.

Text

Description automatically generated

U Fajlu Oprema.jsx importovali smo OpremaKartica i prvo proverili da li je korisnik ulogovan kako bi pristupio stranici a zatim i omogućili sortiranje proizvoda rastuće i opadajuće po ceni proizvoda.

Text

Description automatically generated

Mapiranjem smo omogućili prikaz proizvoda i ikonica za dodavanje i izbacivanje iz korpe. Dok smo u App.js -u dodali konstantu koja preuzima sve proizvode iz baze uz pomoć rute za proizvod i funkcije GET. Takođe iskoristili smo useState kako bismo inicijalizovali početne vrednosti i kasnije ih menjali. Text

Description automatically generated

Sledeći korak je upravo kreiranje korpe, gde smo na početku dodali samo mogućnost prikaza proizvoda, a kasnije dodali još neke funkcionalnosti.

A picture containing text

Description automatically generated

Takođe u App.js-u kreirana je funkcija za dodavanje proizvoda u korpu i osvežavanje korpe, kako bi se menjala i cena proizvoda.

A screenshot of a computer

Description automatically generated with medium confidence

Zatim je dodata i funkcija za brisanje proizvoda iz korpe.

Text

Description automatically generated

Na sledećoj slici prikazane su sve kreirane rute do sad.

Text

Description automatically generated

Zatim smo u fajlu Korpa.jsx omogućili generisanje računa u pdf formatu, za šta je bilo potrebno da instaliramo jspdf i jspdf-autotable. Takođe uveli smo opciju izbora valute RSD,USD i EUR, gde uz pomoć javnog servisa za konvertovanje valuta, konvertujemo u valutu koju je korisnik odabrao iz padajućeg menija.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

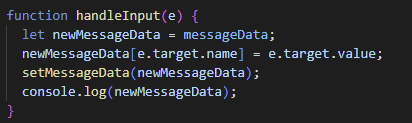
Pored dosada navednih ruta, u App.js tu je još i ruta za komponentu Kontakt kao i rute koje su namenjene adminu prema odgovarajućim komponentama:



Komponenta Kontakt sadrži formu za unos poruke i tri mape:

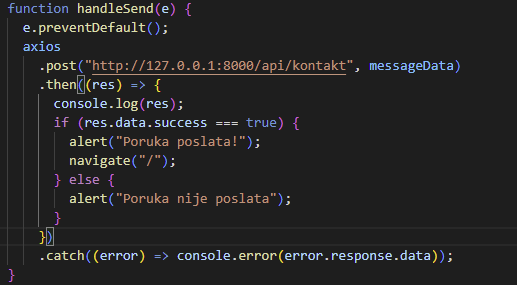


Forma ima tri polja za unos koja prilikom unosa zovu funkciju handleInput():

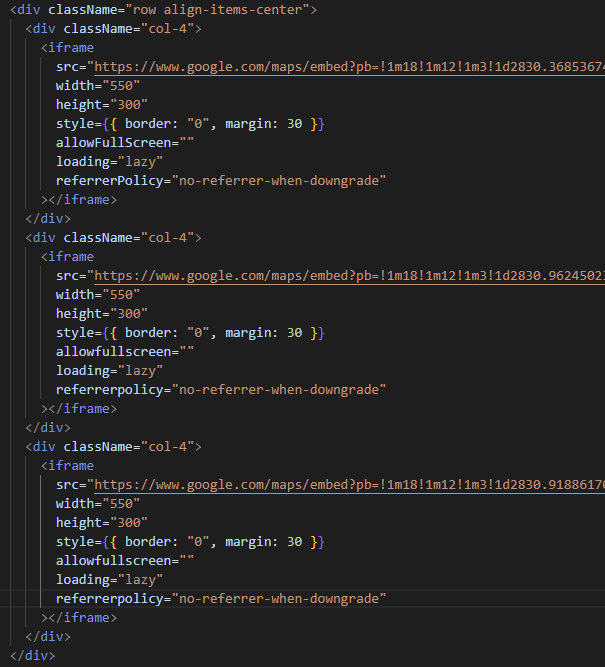


Ova funkcija prima event unosa i postavlja odgovarajuću vrednost konstante messageData na unetu vrednost.

Klikom na dugme „Pošalji“ zove se funkcija handleSend():



Ova funkcija korišćenjem axios-a zove odgovarajuću rutu za čuvanje poruke u bazi i ako je čuvanje uspešno prikazuje poruku o uspešnosti i prebacuje korisnika na početnu stranicu, a u suprotnom ispisuje poruku da poruka nije poslata.



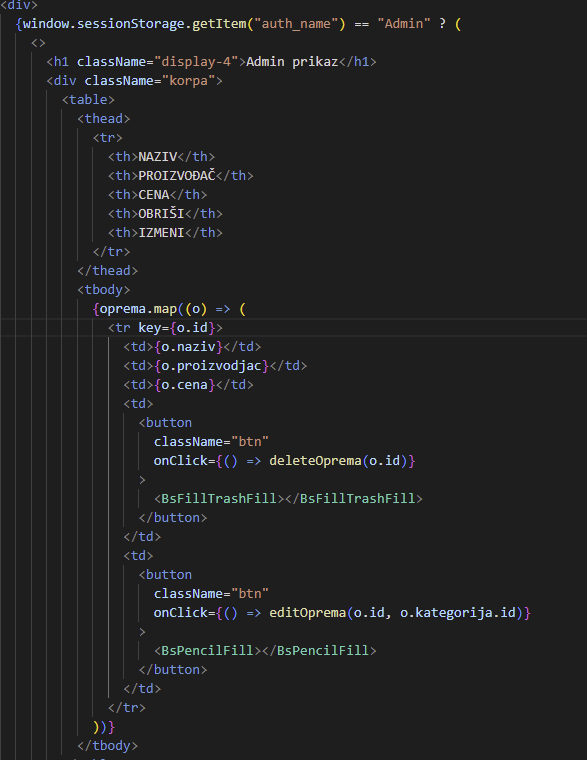
Što se tiče mapa, one su dobijene zvanjem Google Maps API-a, dok su pored toga podešene i njihove veličine i margine.

Komponenta Inbox sadrži tabelu sa porukama koje su poslali posetioci prodavnice:

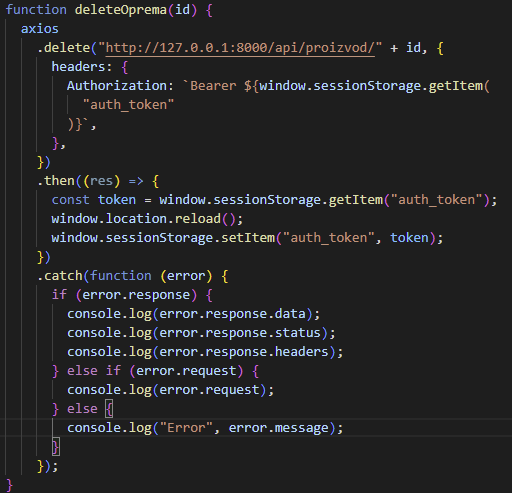


Prvo se u sessionStorage-u proverava da li je ulogovani korisnik admin i ukoliko jeste prikazuje se tabela pomoću komponente MDBDataTableV5 iz paketa mdbreact, a u suprotnom ako ulogovani korisnik nije admin, ispisuje se poruka „Nemate pristup“.

Komponenta AdminView prvo proverava da li je ulogovani korisnik admin i ako nije prikazuje poruku „Nemate pristup“, a u suprotnom prikazuje tabelu sa svom opremom koju preko props-a dobija od App.js kao i dugmiće za brisanje i izmenu opreme:

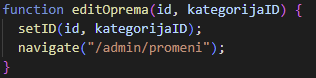


Klikom na dugme za brisanje zove se funkcija deleteOprema iz App.js:



Ova funkcija pomoću axios-a zovu odgovarajuću rutu za brisanje proizvoda i ukoliko je brisanje uspešno refresh-uje stranicu, dok u suprotnom vrši proveru i kontrolu greške koju ispisuje u odgovarajućem obliku.

Klikom na dugme za izmenu zove se funkcija editOprema koja prvo zove funkciju setID koja postavlja vrednost konstante ID u App.js-u na id izabranog proizvoda, a zatim prebacuje korisnika na /admin/promeni stranicu:



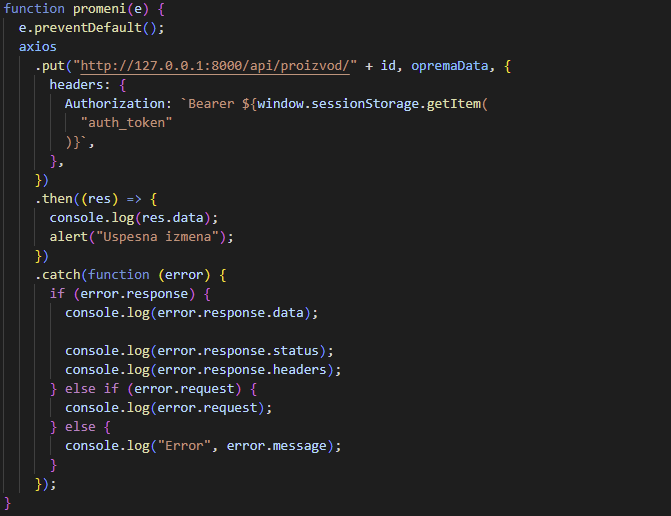
Komponenta Promena takođe proverava da li je ulogovani korisnik admin, i ako nije prikazuje se poruka „Nemate pristup“. Ako jeste, prikazuje se forma sa podacima izabranog proizvoda uz mogućnost njihove izmene:



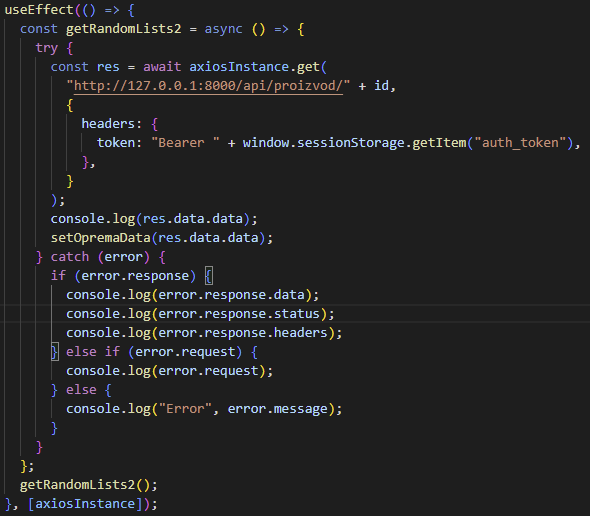




Prilikom izmene bilo kog polja forme, uneta vrednost se čuva u izabranom proizvodu, kao i u Kontakt komponenti, a klikom na dugme „Potvrdi“ zove se funkcija promeni:



Ova fukcija korišćenjem axios-a zove rutu za izmenu odabranog proizvoda i u slučaju uspešne izmene prikazuje se poruka o uspesnosti, u suprotnom, u konzoli se ispisuje odgovarajuća greška.



Pored ovoga, u ovoj komponenti za učitavanje izabranog proizvoda i postavljanje dobijene vrednosti na vrednost konstante opremaData, koristi hook useEffect i axios.

Komponenta Analitika proverava da li ulogovani korisnik admin, i ako nije prikazuje poruku „Nemate pristup“, u suprotnom, prikazuje Plot komponentu:



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 proizvoda, a na y osi njihove cene:

