Fakultet Organizacionih Nauka,

Univerzitet u Beogradu 2021.

Internet Tehnologije.

Veb prodavnica Amazon

Mentor: Student:

prof. dr. Dušan Barać Miloš Belić 29/17

# Sadržaj

[Sadržaj 2](#_Toc80710395)

[Uvod 5](#_Toc80710396)

[Opis problema 5](#_Toc80710397)

[Kategorije korisnika 5](#_Toc80710398)

[Kupac 5](#_Toc80710399)

[Administrator 5](#_Toc80710400)

[Opis proizvoda 5](#_Toc80710401)

[Arhitektura sistema 5](#_Toc80710402)

[Funkcionalni zahtevi 6](#_Toc80710403)

[Funkcionalnosti korisnika (Kupac) stranice 6](#_Toc80710404)

[Funkcionalnosti administratora stranice 6](#_Toc80710405)

[Autorizacija korisnika 7](#_Toc80710406)

[Slučajevi korišćenja 8](#_Toc80710407)

[Scenario za prijavljivanje korisnika 8](#_Toc80710408)

[Scenario pretrage proizvoda 8](#_Toc80710409)

[Scenario za dodavanje proizvoda u korpu 9](#_Toc80710410)

[Scenario za naručivanje proizvoda u korpi 9](#_Toc80710411)

[Scenario uklanjanja proizvoda 10](#_Toc80710412)

[Arhitektura aplikacije 11](#_Toc80710413)

[Opis procesa 11](#_Toc80710414)

[DS: dodavanje proizvoda u korpu 11](#_Toc80710415)

[DS: registracija kupca 12](#_Toc80710416)

[DS: naručivanje proizvoda u korpi 13](#_Toc80710417)

[DS: Uklanjanje proizvoda 14](#_Toc80710418)

[14](#_Toc80710419)

[Model podataka 15](#_Toc80710420)

[Dijagram klasa 15](#_Toc80710421)

[Relacioni model 15](#_Toc80710422)

[Specifikacija API-a 16](#_Toc80710423)

[Registracija 16](#_Toc80710424)

[Login 16](#_Toc80710425)

[Add\_To\_Cart 17](#_Toc80710426)

[CartList 17](#_Toc80710427)

[GetProducts 18](#_Toc80710428)

[Remove\_From\_Cart 18](#_Toc80710429)

[Remove\_From\_Orders 19](#_Toc80710430)

[Search 19](#_Toc80710431)

[GetOrders 20](#_Toc80710432)

[GetAllOrders 20](#_Toc80710433)

[GetAllProducts 20](#_Toc80710434)

[PlaceOrder 21](#_Toc80710435)

[UpdateOrder 21](#_Toc80710436)

[RemoveProduct 22](#_Toc80710437)

[AddProduct 22](#_Toc80710438)

[Opis tehnologija 24](#_Toc80710439)

[Klijentski deo 24](#_Toc80710440)

[Serverski deo 24](#_Toc80710441)

[Baza podataka 24](#_Toc80710442)

# 

# Uvod

Projekat Veb prodavnica Amazon je izrađen u okviru praktične nastave na predmetu ITEH Fakulteta Organizacionih Nauka. Osnovni cilj projekta je kreiranje internet prodavnice koja omogućava pregled i kupovinu proizvoda.

Svrha ovog dokumenta je da služi kao dokumentacija za projekat i opis slučajeva korišćenja, API-ja itd. Projekat možete naći na repozitorijumu <https://github.com/milos-531/amazon-project>

# Opis problema

Danas je skoro neophodno imati onlajn prisustvo ukoliko imate prodavnicu. Od veb prodavnice se očekuje da može beležiti korisnike, pružiti uvid u dostupne proizvode i omogućiti kupovinu i dostavu proizvoda. Ovaj projekat omogućava korišćenje u upravo te svrhe.

# Kategorije korisnika

U sistemu postoje dve kategorije korisnika: Kupac i administrator.

## Kupac

Kupcu je omogućeno sledeće: pristup proizvodima, pretraga proizvoda, dodavanje proizvoda u korpu, naručivanje proizvoda iz korpe, prijavljivanje i registracija.

## Administrator

Administratoru je omogućeno sledeće: pregled proizvoda i narudžbina, obeležavanje narudžbina u odnosu na status plaćanja i status dostave, uklanjanje proizvoda i dodavanje novih proizvoda.

# Opis proizvoda

U ovom odeljku je dat opis projekta sa najvišeg nivoa, uz tehnologije korišćene.

## Arhitektura sistema

Možemo reći da se sistem sastoji od sajta na klijentskoj strani realizovanog putem React tehnologije, serverske strane koja obrađuje zahteve, implementirane u Laravelu, i MySql baze podataka za skladištenje podataka.

# Funkcionalni zahtevi

U ovoj sekciji se navode svi zahtevi koji moraju biti ispunjeni za korisnike svih uloga. Onde gde način prenosa podataka i evidentiranja nije specijalno naznačen, podrazumeva se da se unos vrši preko odgovarajuće React stranice, slanjem podataka ka odgovarajoj ruti API-ja, koji zatim izvrši potrebne promene u bazi podataka.

## Funkcionalnosti korisnika (Kupac) stranice

### Pregled proizvoda

Korisnik ima pristup nekim proizvodima na osnovnoj stranici (homepage).

### Pretraga proizvoda

Korisnik ima mogućnost pretraživanja proizvoda po imenu, opisu i kategoriji.

### Dodavanje u korpu

Ulogovani korisnik ima mogućnost dodavanja proizvoda u svoju korpu

### Uklanjanje iz korpe

Ulogovani korisink ima mogućnost uklanjanja proizvoda iz korpe.

### Naručivanje

Ulogovani korisnik ima mogućnost naručivanja proizvoda iz svoje korpe.

### Uklanjanje narudžbine

Ulogovani korisink ima mogućnost uklanjanja proizvoda iz narudžbina.

## Funkcionalnosti administratora stranice

### Pregled narudžbina

Administrator ima pregled svih narudžbina na sistemu, uz informacije o njihovom statusu.

### Pregled proizvoda

Administrator ima pregled svih proizvoda na sistemu.

### Obeležavanje narudžbina

Administrator ima mogućnost obeležavanja statusa plaćanja i dostave narudžbina.

### Uklanjanje proizvoda

Administrator ima mogućnost uklanjanja proizvoda iz sistema.

### Dodavanje proizvoda

Administrator ima mogućnost dodavanja novih proizvoda u sistem.

## Autorizacija korisnika

Korisnik pristupa svom nalogu unosom email-a i lozinke, koji se moraju poklapati sa odgovarajućim vrednostima u bazi. Omogućena je i registracija Kupac. Nakon uspešnog prijavljivanja, korisnik se prosleđuje odgovarajućoj stranici. Administratori i regularni korisnici se prijavljuju putem iste stranice.

# Slučajevi korišćenja

## Scenario za prijavljivanje korisnika

### Kratak opis

Vrši se putem unosa korisničkog imena i lozinke. Uneti podaci se moraju poklapati sa podacima iz baze podataka. Nakon uspešne prijave, korisnik može da interaguje sa sistemom i koristi usluge koje su mu dozvoljene.

### Tok događaja

1. Korisnik unosi korisničko ime i lozinku i bira opciju da se uloguje
2. Sistem poredi unete podatke sa onima iz baze
3. Sistem prikazuje početnu stranicu i korisnik je ulogovan i ima odredjene privilegije

3.1 Ukoliko se podaci ne poklapaju sa onima iz baze sistem će ispisati obaveštenje da autorizacija nije uspela i dolazi do povratka na korak 1.

### Posebni zahtevi

Nema

### Preduslovi

Da bi autorizacija bila uspešna potrebno je da nalog postoji u bazi podataka.

### Posledice

Korisnik je ulogovan u sistem.

## Scenario pretrage proizvoda

### Kratak opis

Korisnici imaju mogućnost da pretraže sve proizvode koji se nalaze u bazi. Korisnik ima mogućnost sortiranja vraćenih proizvoda po ceni, kao i nazivu.

### Tok događaja

1. Korisnik unosi vrednost za pretragu i bira opciju za pretragu
2. Sistem prikazuje proizvode u tabeli
3. Korisnik bira opcije za sortiranje proizvoda
4. Sistem sortira listu vraćenih proizvoda po odgovarajućoj vrednosti

3.1 Ako ne postoji nijedan proizvod koji odgovara pretrazi, biće prikazana prazna lista

### Posebni zahtevi

Nema

### Preduslovi

Nema

### Posledice

Nema

## Scenario za dodavanje proizvoda u korpu

### Kratak opis

Kupac ima mogućnost dodavanja proizvoda koje vidi u korpu. Nakon odabira opcije, dodaje se vrednost u odgovarajuću tabelu.

### Tok događaja

1. Korisnik bira opciju za dodavanje proizvoda u korpu
2. Sistem ažurira broj proizvoda u korpi, prikazan u navbaru

2.1 Ukoliko dođe do greške, korisniku se ispisuje odgovarajuća poruka

### Posebni zahtevi

Nema

### Preduslovi

Korisnik je ulogovan kao Kupac

### Posledice

Dodaje se proizvod u odgovarajuću tabelu u bazi podataka

## Scenario za naručivanje proizvoda u korpi

### Kratak opis

Kupac ima mogućnost naručivanja proizvoda koji se nalaze u korpi. Nakon odabira opcije, vodi se do forme za popunjavanje potrebnih informacija.

### Tok događaja

1. Korisnik bira opciju za naručivanje proizvoda u korpi
2. Sistem prikazuje stranicu za naručivanje (Confirm order)
3. Korisnik popunjava formu i bira opciju za naručivanje
4. Sistem prikazuje da je uspešno poslat zahtev i prosleđuje korisnika do stranice sa narudžbinama

4.1 Ukoliko je neko polje neispravno uneseno, sistem će prikazati poruku o grešci i dolazi do povratka na korak 5. Ukoliko je došlo do greške pri obradi zahteva, korisnk će videti odgovarajuću poruku

### Posebni zahtevi

Nema

### Preduslovi

Korisnik je ulogovan kao Kupac

### Posledice

Dodaju se narudžbine u odgovarajuću tabelu u bazi podataka

## Scenario uklanjanja proizvoda

### Kratak opis

Administrator ima mogućnost uklanjanaj proizvoda koji trenutno ne čeka na dostavu

### Tok događaja

1. Administrator bira opciju uklanjanja proizvoda
2. Sistem uklanja proizvod i ažurira prikaz svih proizvoda koje administrator vidi

2.1 Ako proizvod postoji u narudžbini koja očekuje dostavu (nije poništena ili dostavljena), uklanjanje se prekida i administratoru se prikazuje odgovarajuća poruka

### Posebni zahtevi

Nema

### Preduslovi

Korisnik je ulogovan kao administrator. Postoje proizvodi u bazi

### Posledice

Proizvod je uklonjen iz odgovarajućih tabela

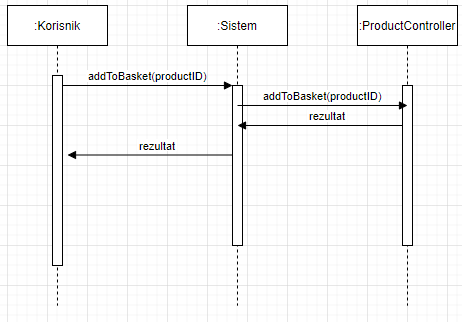
# Arhitektura aplikacije

Aplikacija se sastoji iz korisničkog (front) dela i dela za obradu zahteva (back). Aplikacija je tipa klijent-server gde klijentska strana omogućava pristup veb prodavnici i svim funkcionalnostima. Klijentska strana se sastoji iz više veb stranica različitih mogućnosti. Klijentska strana komunicira sa serverskom slanjem HTTP zahteva ka serveru, i razvijen je poseban API za obradu zahteva. Serverska strana sadrži mnoštvo ruta koje prosleđuju zahteve do kontrolera. Serverska strana takođe sadrži alatke za rad sa bazom podataka, što omogućava serverskim kontrolerima pristup bazi podataka. Serverska strana takođe sadrži migracije i seeder-e koji omogućavaju kreiranje odgovarajućih tabela i punjenje istih početnim vrednostima.

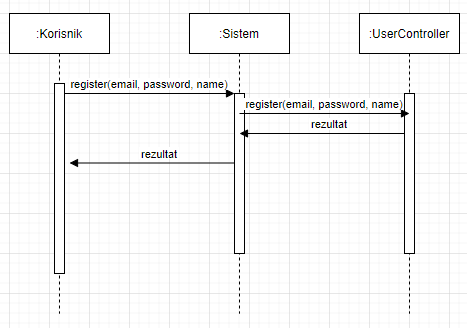
# Opis procesa

Dijagrami sekvenci prikazuju interakciju između klijentskih i serverskih klasa i funkcija, izostavljaju se pozivi ka API-ju koji se vrše između jer će se oni detaljno opisati nadalje

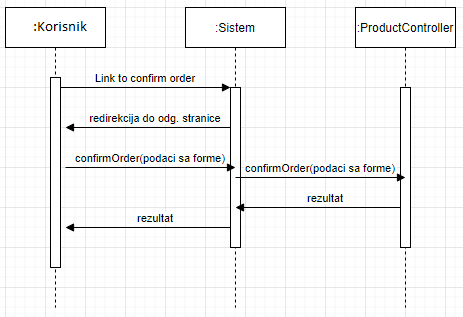
## DS: dodavanje proizvoda u korpu



## DS: registracija kupca



## DS: naručivanje proizvoda u korpi

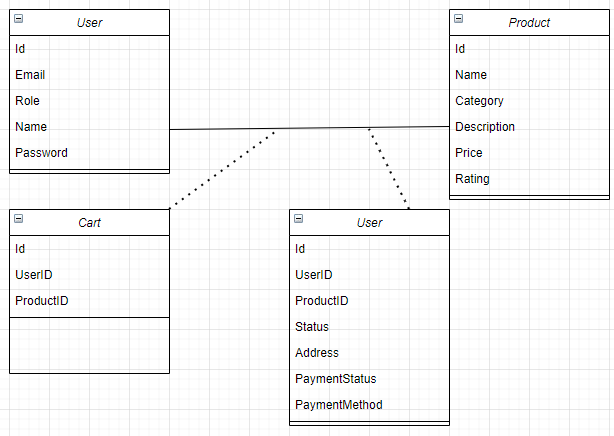


## DS: Uklanjanje proizvoda

## 

# Model podataka

## Dijagram klasa



## Relacioni model

Users(Id, Name, Role, Email, Password)

Products(Id, Name, Category, Description, Price, Rating)

Cart(Id, *User\_id*, *Product\_id*)

Orders(Id, *User\_id*, *Product\_id*, Address, Status, Payment\_Status, Payment\_method)

# Specifikacija API-a

## Registracija

### Opis:

Registruje novog kupca

### Endpoint:

http://localhost:8000/api/register

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {name, email, password} (svi parametri stringovi)

Body: item

Format: JSON

### Rezultat:

0 – Email je već zauzet

User - JSON objekat koji predstavlja novokreirani red u tabeli korisnika

## Login

### Opis:

Prijavljuje korisnika koji već postoji u bazi

### Endpoint:

http://localhost:8000/api/login

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {email, password} (svi parametri stringovi)

Body: item

Format: JSON

### Rezultat:

0 – Email nije pronađen, ili neispravna lozinka

User - JSON objekat koji predstavlja odgovarajući red iz tabele korisnika

## Add\_To\_Cart

### Opis:

Dodaje proizvod u korpu prijavljenog kupca

### Endpoint:

### http://localhost:8000/api/add\_to\_cart

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {user\_id, product\_id} (svi parametri celi brojevi)

Body: item

Format: JSON

### Rezultat:

0 – Greška prilikom dodavanja u korpu

Cart - JSON objekat koji predstavlja novokreirani red u tabeli korpe

## CartList

### Opis:

Vraća listu svih proizvoda u korpi korisnika

### Endpoint:

### http://localhost:8000/api/cartlist/ + id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

id – Šifra ulogovanog korisnika, prosleđena putem URL-a

### Rezultat:

Products - JSON objekat koji predstavlja sve proizvode u tabeli korpe za datog korisnika

## GetProducts

### Opis:

Vraća listu proizvoda date veličine

### Endpoint:

### http://localhost:8000/api/getproducts/ + id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

id – Broj proizvoda za vraćanje, prosleđeno putem URL-a

### Rezultat:

Products - JSON objekat koji predstavlja listu redova iz tabele proizvoda tačno {id} veličine

## Remove\_From\_Cart

### Opis:

Uklanja proizvod iz korpe za datog kupca

### Endpoint:

### http://localhost:8000/api/remove\_from\_cart

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {user\_id, product\_id} (svi parametri celi brojevi)

Body: item

Format: JSON

### Rezultat:

Throwable th – Greška prilikom uklanjanja iz korpe

1 – Uspeh prilikom uklanjanja iz korpe

## Remove\_From\_Orders

### Opis:

Uklanja proizvod iz narudžbina za datog kupca

### Endpoint:

### http://localhost:8000/api/remove\_from\_orders

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {user\_id, product\_id} (svi parametri celi brojevi)

Body: item

Format: JSON

### Rezultat:

0 – Greška prilikom uklanjanja iz korpe

1 – Uspeh prilikom uklanjanja iz korpe

## Search

### Opis:

Vraća proizvode kojima ime, kategorija ili opis sadrže odgovarajući tekst

### Endpoint:

### http://localhost:8000/api/search/ + query

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

query– Tekst za pretragu, prosleđeno putem URL-a

### Rezultat:

Products - JSON objekat koji predstavlja listu redova iz tabele proizvoda koji odgovaraju pretrazi

## GetOrders

### Opis:

Vraća sve proizvode za datog kupca

### Endpoint:

### http://localhost:8000/api/getorders/ + user\_id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

User\_id– Šifra kupca

### Rezultat:

Orders - JSON objekat koji predstavlja listu redova iz tabele narudžbina koji odgovaraju kupcu

## GetAllOrders

### Opis:

Vraća sve narudžbine ukoliko je korisnik administrator

### Endpoint:

### http://localhost:8000/api/getallorders/ + user\_id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

User\_id– Šifra administratora

### Rezultat:

-1 – Ukoliko korisnik nije administrator

Orders - JSON objekat koji predstavlja listu svih redova iz tabele

## GetAllProducts

### Opis:

Vraća sve proizvode ukoliko je korisnik administrator

### Endpoint:

### http://localhost:8000/api/getallproducts/ + user\_id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

User\_id– Šifra administratora

### Rezultat:

-1 – Ukoliko korisnik nije administrator

Products - JSON objekat koji predstavlja listu svih redova iz tabele

## PlaceOrder

### Opis:

Kreira narudžbinu na osnovu korpe kupca

### Endpoint:

### http://localhost:8000/api/placeorder

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = {user\_id, address, pazment} (int, string, string)

Body: item

Format: JSON

### Rezultat:

0 – Greška prilikom dodavanja u korpu

1 – Uspeh prilikom dodavanja u korpu

## UpdateOrder

### Opis:

Ažurira status dostave ili kupovine narudžbine

### Endpoint:

### http://localhost:8000/api/updateorder

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = JSON objekat kreiran na osnovu reda iz tabele narudžbina

Body: item

Format: JSON

### Rezultat:

Throwable th – Greška prilikom dodavanja u korpu

orders - JSON lista narudžbina

## RemoveProduct

### Opis:

Uklanja proizvod iz tabele

### Endpoint:

### http://localhost:8000/api/removeproduct/ + id

### Metoda:

GET

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

id – Broj proizvoda za uklanjanje, prosleđeno putem URL-a

### Rezultat:

Products - JSON objekat koji predstavlja listu svih redova iz tabele proizvoda

## AddProduct

### Opis:

Dodaje novi proizvod

### Endpoint:

### http://localhost:8000/api/addproduct

### Metoda:

POST

### Zaglavlja:

"Content-Type": "application/json",

"Accept": "application/json"

### Parametri:

item = JSON objekat sa svim promenljivima proizvoda

Body: item

Format: JSON

### Rezultat:

-1 – Greška prilikom dodavanja proizvoda

-2- Greška postoji proizvod istog imena

# Opis tehnologija

## Klijentski deo

Korisnički deo je izrađen uz pomoć React tehnologije. Organizovanje različitih stranica je izvršeno uz pomoć React Router alatke, uz koju su kreirane putanje u aplikaciji. Ikonice poput onih za pretragu i sortiranje su dobijene od material-ui icons biblioteke. Korišćene su i funkcionalne i klasne komponente. Skladištenje podataka na stranici je izvedeno uz pomoć React Context API-a. Korišćeno je mnoštvo komponenti poput proizvoda koje se mogu ponavljati više puta. Korišćen je javni veb servis za validaciju emailova prilikom registracije.

## Serverski deo

Serverski deo je izrađen uz pomoć Laravela. Sastoji se od ruta koje zahteve prosleđuju do odgovarajućih kontrolera, koji su u mogućnosti da pristupe bazi podataka, kao i da vrše izmene nad istom. Laravel modeli štite protiv SQL injekcije, osim u slučaju RAW funkcija. Jedna takva funkcija je korišćena samo u slučaju pretrage proizvoda, i tada je korišćen parametrizovani upit. Dakle SQL injekcije nisu pretnja u aplikaciji.

## Baza podataka

Korišćen je MySql sistem za skladištenje i obradu podataka.