

# Moderni pristup izradi web aplikacije

---

**Puhanić, Roberto**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Tourism and Rural Development in Pozega / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet turizma i ruralnog razvoja u Požegi**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:277:039381>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-23**



*Repository / Repozitorij:*

[FTRR Repository - Repository of Faculty Tourism and Rural Development Pozega](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET TURIZMA I RURALNOG RAZVOJA U POŽEGI**



**Robertino Puhanić, 0253054638**

**Moderan pristup izrade web aplikacije**

**ZAVRŠNI RAD**

Požega, 2024. godine.

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET TURIZMA I RURALNOG RAZVOJA U POŽEGI**

**PRIJEDIPLOMSKI STUDIJ ELEKTRONIČKO POSLOVANJE I PROGRAMSKO  
INŽENJERSTVO**

**Moderan pristup izrade web aplikacije  
ZAVRŠNI RAD**

**IZ KOLEGIJA PROGRAMIRANJE WEB APLIKACIJA II**

**MENTOR: prof. dr. sc. Krešimir Nenadić**

**STUDENT: Robertino Puhanić**

**JMBAG studenta: 0253054638**

**Požega, rujan 2024. godine**

## SAŽETAK

Web trgovina „Print Alfa“ izrađena je u svrhu demonstracije kako koristiti moderne alate za izradu raznih web aplikacija. Aplikacija je podijeljena na dva glavna dijela, na poslužiteljsku i klijentsku stranu. Na poslužiteljskoj strani se opisuje kompletan proces instalacije radnog okruženja s Next.js, povezivanje s MongoDB Atlasom odnosno bazom podataka, autentifikacija putem kontrolne ploče, kreirane su API rute za upravljanje proizvodima te modeliranje podataka u bazi podataka.

Aplikacija posjeduje funkcionalnosti poput prijave unutar kontrole ploče koja je omogućena samo administratorima, dodavanje proizvoda, dodavanje cijena, opisa, kategorija te fotografija, izradu novih kategorija te potkategorija, brisanje i uređivanje proizvoda, te pregled svih narudžbi. Uz pomoć Tailwind CSS te njegovih paketa aplikacija postiže responzivan i privlačan dizajn. Prikazan je način manipulacije košarice uz pomoć kuka 'useState', 'useEffect' i 'createContext'. Navigacija je napravljena pomoću 'pages' mape u koju se stavljaju sve podstranice.

Web trgovina „Print Alfa“ služi kao primjer kako se uz pomoć modernih tehnologija poput Next.js, React i Tailwind CSS mogu izgraditi responzivne, funkcionalne i moderne web aplikacije koje zadovoljavaju norme modernog web razvoja.

**Ključne riječi:** Next.js, React, Tailwind CSS, MongoDB, API, pages

## **ABSTRACT**

The "Print Alfa" web store was created for the purpose of demonstrating how to use modern tools for creating various web applications. The application is divided into two main parts, the server side and the client side. On the server side, the complete process of installing the working environment with Next.js, connecting to MongoDB Atlas or the database, authentication through the control panel, API routes for product management and data modeling in the database are described.

The application has functionalities such as logging into the control panel, which is only available to administrators, adding products, adding prices, descriptions, categories and photos, creating new categories and subcategories, deleting and editing products, and viewing all orders. With the help of Tailwind CSS and its application packages, it achieves a responsive and attractive design. The way to manipulate the shopping cart with the help of the 'useState', 'useEffect' and 'createContext' hooks is shown. Navigation is made using the 'pages' folder, where all subpages are placed.

The "Print Alfa" web store serves as an example of how, with the help of modern technologies such as Next.js, React and Tailwind CSS, responsive, functional and modern web applications can be built that meet the standards of modern web development.

**Keywords:** Next.js, React, Tailwind CSS, MongoDB Atlas, API, pages

## Sadržaj

1. UVOD .....	1
2. PREGLED SLIČNIH APLIKACIJA .....	2
2.1 Korisničko sučelje i korisničko iskustvo (UI/UX) .....	2
2.2. Funkcionalnost i tehnološke značajke .....	5
2.3. Sigurnost i privatnost korisnika.....	6
3. FUNKCIONALNI ZAHTJEVI APLIKACIJE .....	7
4. KORIŠTENE TEHNOLOGIJE.....	8
4.1. Next.js .....	8
4.2. React.js.....	9
4.3. MongoDB.....	9
4.4. Tailwind CSS .....	9
5. POSTUPAK IZRADA APLIKACIJE .....	10
5.1. Opis izrade poslužiteljskih funkcionalnosti.....	10
5.1.1. Povezivanje s MongoDB Atlas .....	11
5.1.2. Autentikacija s kontrolnom pločom .....	12
5.1.3. API ruta za upravljanje proizvodima.....	12
5.1.4. Model za upravljanje podacima u bazi podataka.....	13
5.2. Opis izrade klijentske strane .....	14
5.2.1. Izrada komponenti.....	14
5.2.2. Komponenta za navigaciju.....	15
5.2.3. Direktorij pages.....	15
5.2.4. React Hooks .....	16
5.2.5 Logika dodavanja i brisanja proizvoda unutar košarice .....	17
5.2.6 Stripe integracija .....	18
6. PRIKAZ RADA APLIKACIJE .....	19
6.1. Prikaz rada aplikacije – klijentska strana .....	19
7. ZAKLJUČAK .....	24
LITERATURA.....	25
POPIS SLIKA.....	26

## 1. UVOD

Razvoj web aplikacija danas zahtijeva upotrebu suvremenih tehnologija koje omogućuju visoku efikasnost, sigurnost i ugodno korisničko iskustvo. Ovaj rad se bavi izradom moderne web trgovine koristeći Next.js, React.js, Tailwind CSS i MongoDB, sa svrhom kreiranja cjelovite aplikacije koja objedinjuje poslužiteljski i klijentski dio.

Web trgovina je strukturirana tako da poslužiteljski dio obuhvaća upravljanje podacima, autentikaciju korisnika i sigurno povezivanje s bazom podataka korištenjem MongoDB Atlas. Poslužiteljski dio također uključuje implementaciju API (engl. *Application Programming Interface*) ruta za obavljanje ključnih CRUD (eng. Create, Read, Update and Delete) operacija nad proizvodima, što administratorima omogućava dodavanje, uređivanje i brisanje proizvoda te upravljanje kategorijama.

Klijentski dio aplikacije fokusira se na kreiranje korisničkog sučelja pomoću React komponenti, omogućavajući modularnost i ponovnu upotrebu koda. Navigacija između stranica ostvarena je uz pomoć Next.js direktorija *pages*, što olakšava razvoj i održavanje aplikacije. Posebna pažnja posvećena je upravljanju stanjem aplikacije korištenjem funkcionalnosti React Hooks, kao što su *useState* i *useEffect*, što omogućava dinamično i responzivno korisničko iskustvo. Dodatno, logika upravljanja košaricom implementirana je putem *CartContext* komponente koja osigurava jednostavno dodavanje i uklanjanje proizvoda iz košarice.

Sigurnost i pouzdanost aplikacije pojačane su integracijom NextAuth za autentikaciju korisnika, omogućujući prijavu putem raznih pružatelja usluga poput Googlea, čime se osigurava da samo ovlašteni korisnici imaju pristup osjetljivim funkcijama aplikacije. Integracija sa Stripe servisom za naplatu omogućuje korisnicima sigurno i jednostavno obavljanje transakcija koristeći različite načine plaćanja, poput kreditnih kartica i digitalnih novčanika.

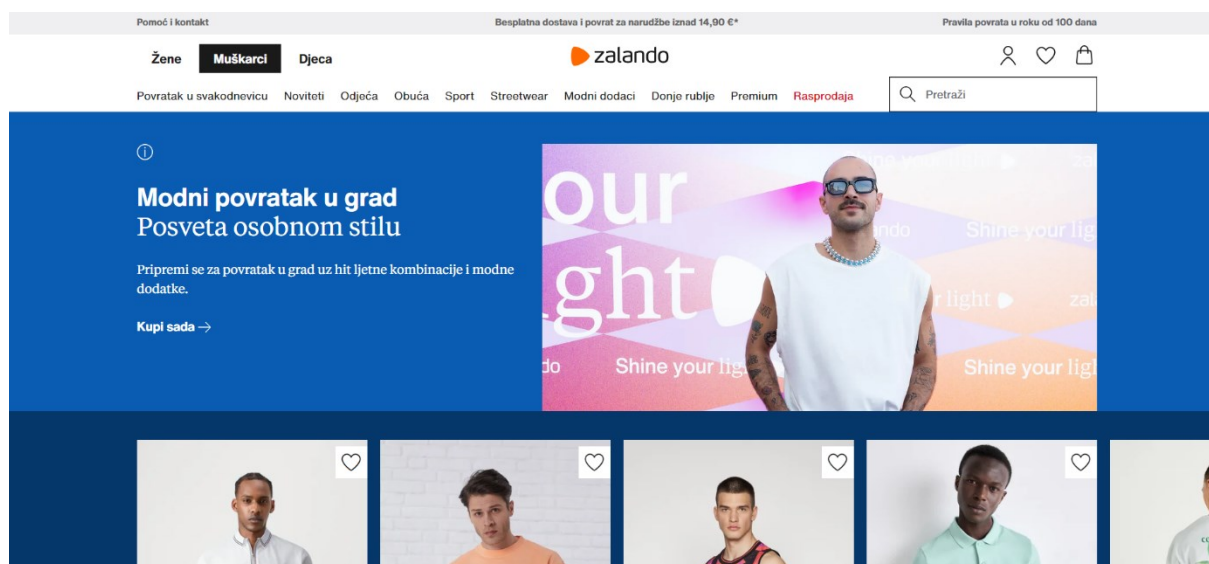
Na kraju prikazan je rad aplikacije u praksi, uključujući demonstraciju glavnih značajki i korisničkog iskustva. Pregled pruža sveobuhvatan uvid u razvoj aplikacije od analize i planiranje do implementacije.

## 2. PREGLED SLIČNIH APLIKACIJA

U digitalnom dobu, online trgovine predstavljaju važan kanal za kupovinu i prodaju proizvoda diljem svijeta. Popularne web trgovine kao što su Apple, Amazon, Ebay, Ikea i Zalando postavili su standarde u pogledu korisničkog iskustva, funkcionalnosti, sigurnosti i tehnoloških inovacija. Cilj ovog poglavlja je usporediti izrađenu web trgovinu s navedenim platformama kako bi se istaknule sličnosti, razlike i specifične prednosti koje ostale web trgovine imaju u odnosu na izrađenu web trgovinu. Analiza se temelji na sljedećim područjima: korisničko sučelje i korisničko iskustvo UI/UX (eng. User Interface / User Experience), funkcionalnosti i tehnološke značajke, sigurnost i privatnost korisnika.

### 2.1 Korisničko sučelje i korisničko iskustvo (UI/UX)

Zalando je fokusiran na modnu industriju, koristi minimalistički dizajn s bijelom pozadinom i velikim fotografijama proizvoda. Izbornici su jednostavni za korištenje s istaknutim kategorijama za odjeću, obuću i dodatke. Koristi i filtre i pretrage po veličinama, bojama, brendovima, popustima, cijenama. Zalando pruža personalizirano iskustvo kupovine, s preporukama proizvoda temeljenim na povijesti pretraživanja i preferencijama korisnika. Dizajn za mobilne uređaje (pametne telefone i tablete) je također optimiziran, stranica se brzo učitava te ima fotografije prilagođene za manje ekrane.



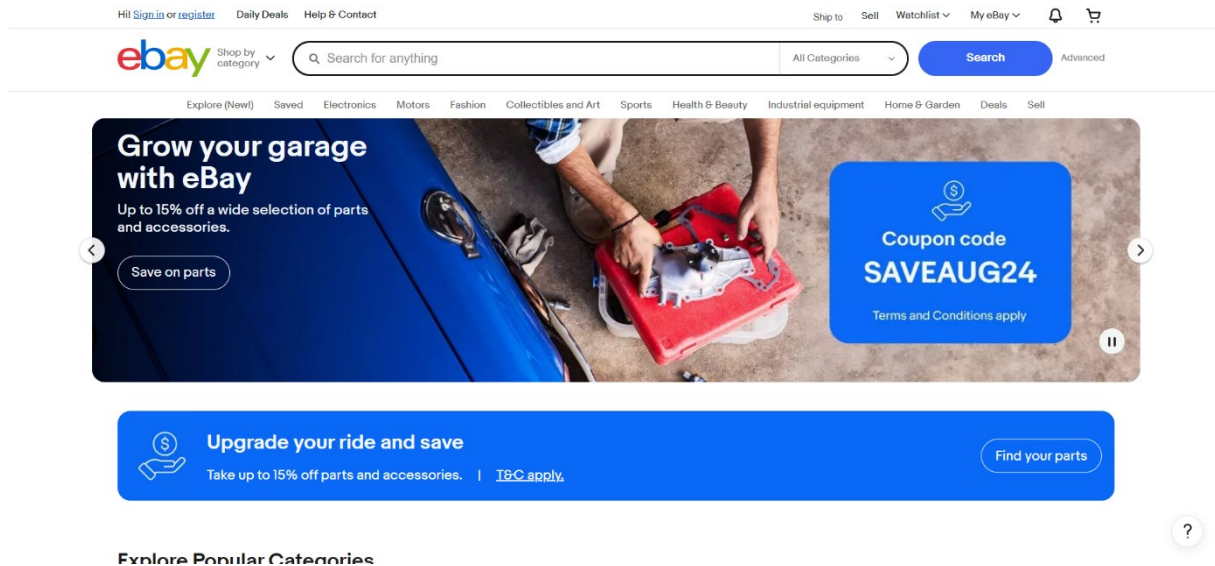
Slika 1 - Zalando web trgovina

Izvor: <https://www.zalando.hr/muskarci-home/>

Ebay koristi jednostavno korisničko sučelje. Dizajn web trgovine je nešto staromodniji u odnosu na ostale web trgovine, a naglasak je na funkcionalnost i dostupnost informacija. Korisničko sučelje je ispunjeno elementima kao što su kategorije, oglasi, ponude te promotivni



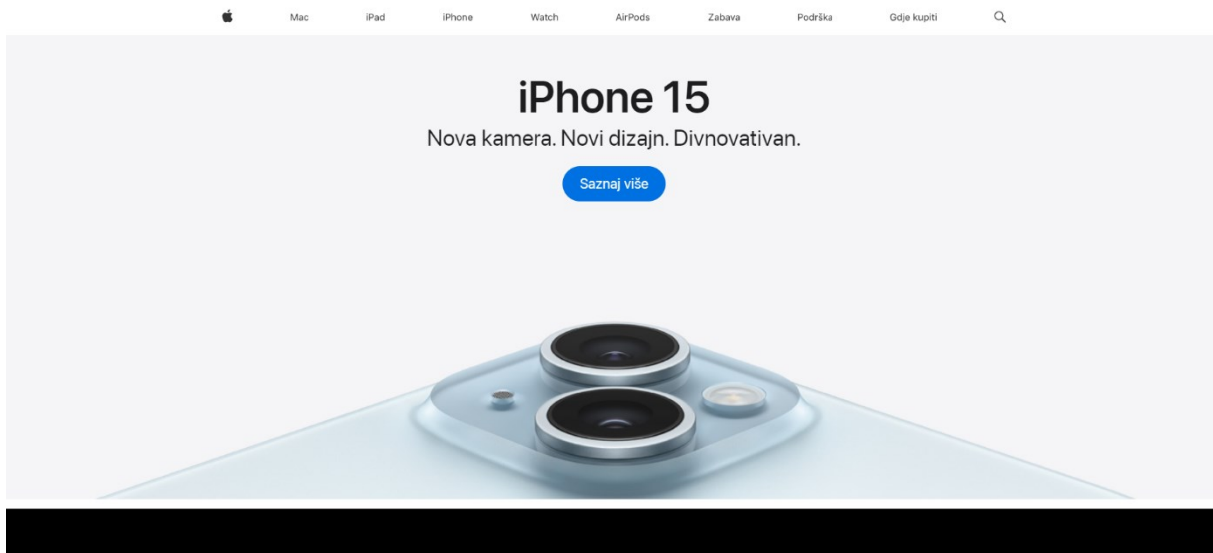
sadržaji. Stranica ima veći broj tekstualnih informacija. Ebay je fokusiran na pretraživanje i kategorizaciju proizvoda. Korisnici mogu s lakoćom pratiti ponude, prodati vlastite proizvode, te koristiti razne metode plaćanja. Web trgovina je prilagođena korisnicima koji traže specifične proizvode te ih žele kupiti po pristupačnijim cijenama.



Slika 2 - Ebay web trgovina

Izvor: <https://www.ebay.com/>

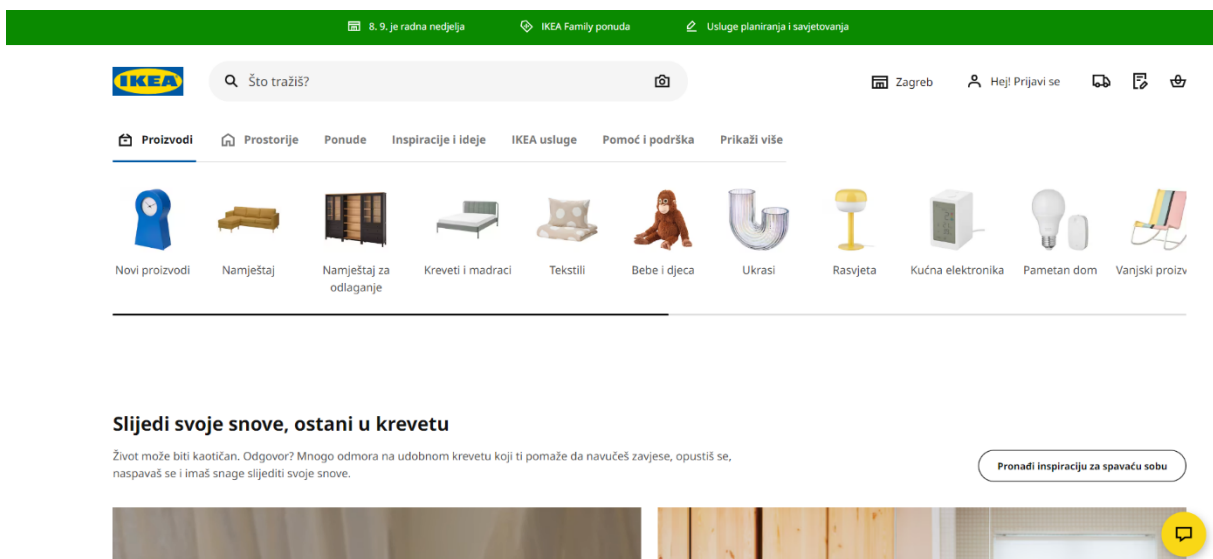
Web trgovina tvrtke Apple je poznata po svom luksuznom i minimalističkom dizajnu kakav je i sam brend. Boje su jednostavne, prevladavaju bijela, crna i siva što predstavlja eleganciju. Na web trgovini se nalazi veća količina praznog prostora što dodatno govori o eleganciji i profinjenom dizajnu. Glavni izbornik je smješten u vrhu stranice, što omogućava brz pristup različitim kategorijama proizvoda. Web trgovina ima vrlo brzu responzivnost na korisnikove akcije s optimiziranim učitavanjem sadržaja. Informacije o svim proizvodima su vrlo detaljne, sa specifikacijama i korisničkim recenzijama. Kupovina je vrlo jednostavna jer nije potrebno prebacivanje između većeg broja zaslona s koracima, a integracija s Apple ID-om omogućuje brzu prijavu i plaćanje. Apple koristi proširenu stvarnost odnosno pregled proizvoda u stvarnom svijetu, što je u odnosu na ostale web trgovine jedinstvena značajka.



Slika 3 - Apple web trgovina

Izvor: <https://www.apple.com/hr/>

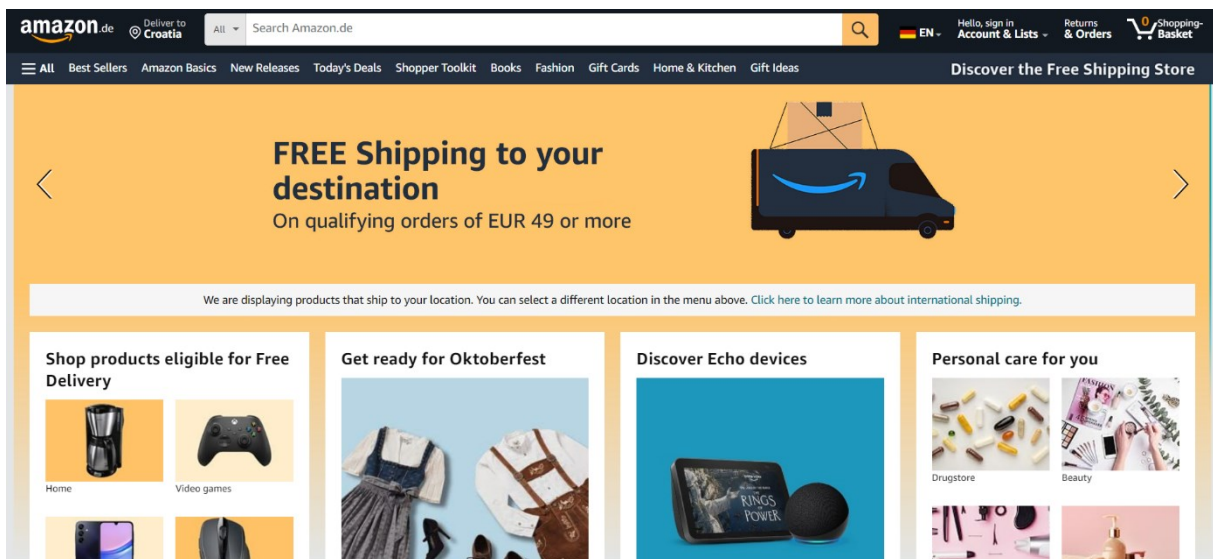
Ikea koristi web trgovinu koja se fokusira na veliki katalog s puno kategorija. Dizajn je praktičan i funkcionalan, a koristi puno vizualnog sadržaja (fotografija proizvoda, videozapisa). Web trgovina koristi neutralne boje (bijela, žuta, plava) koje su u skladu s brendom Ikea. Ikea stavlja fokus na inspiraciju i praktičnost. Unutar web trgovine može se pronaći značajan broj fotografija različitih vrsta prostora (radna soba, dnevni boravak, kuhinja, ili nekih drugih prostorija) te uz pomoć toga pronaći inspiraciju za opremanje doma ili radnog prostora. Ikea svojim korisnicima također nudi online planer koji omogućava dizajniranje vlastitih prostorija korištenjem Ikea namještaja.



Slika 4 - Ikea web trgovina

Izvor: <https://www.ikea.com/hr/hr/>

Amazon koristi funkcionalan dizajn koji se fokusira na dostupnost i učinkovitost. Na stranici se nalazi značajna količina informacija kao što su razni proizvodi, preporuke, oglasi, promotivne ponude. Koristi jednostavne boje (bijela, siva, crna, narančasta). Navigacija je poprilično složena te teška za korištenje, sadrži veći broj izbornika te znatan broj različitih potkategorija. Amazon posjeduje sofisticirane algoritme koji nude preporučene proizvode koji su već pretraživani. Kako bi pronašli određeni proizvod najjednostavnije je koristiti pretraživanje. Također, web trgovina posjeduje napredne opcije praćenja pošiljke što čini iskustvo brzim i efikasnim.



Slika 5 - Amazon web trgovina

Izvor: <https://www.amazon.de/>

Print Alfa web trgovina koristi jednostavno korisničko sučelje. Koristi bijelu pozadinu s crvenim detaljima. Izbornik je jednostavan za korištenje. Klikom na jedan gumb mogu se pregledati svi proizvodi na web trgovini. Sadržaj u web trgovini se učitava vrlo brzo.

## 2.2. Funkcionalnost i tehnološke značajke

Zalando posjeduje napredne mogućnosti pretraživanja i filtriranja koji proizvode mogu pretraživati po različitim kategorijama, a također omogućava korisnicima pregled osnovnih informacija o proizvodu bez otvaranja druge stranice. Na Zalandu se korisnici mogu jednostavno prijaviti i registrirati, također posjeduju registraciju i prijavu putem društvenih mreža kao što je Facebook, a također je moguće obaviti prijavu putem Google računa ili Apple ID-a. Na Zalandu korisnici također imaju mogućnost kreiranja liste želja te praćenja proizvoda odnosno njihove dostupnosti, posjeduju integraciju sa servisima za brzu dostavu pomoću kojih

je moguće brzo i jednostavno pratiti obavljene narudžbe. Zalando posjeduje mobilne aplikacije za iOS i Android.

Ebay nudi mogućnost povrata novca u slučaju problema s narudžbom, posjeduju višestruke načine plaćanja kao što su PayPal, Apple Pay, Google Pay te ima podršku za velik broj kreditnih kartica. Korisnici mogu prodavati vlastite proizvode na Ebayu, a na web trgovini je također moguće ocjenjivati i recenzirati prodavače, što pomaže u izgradnji povjerenja. Ebay također u ponudi ima i mobilne aplikacije za iOS i Android.

Apple omogućava korištenje Apple ID-a za prijavu i plaćanje što omogućava brzu i sigurnu kupovinu. Korisnicima se nude preporuke i podrška koje se temelje na njihovoj povijesti korištenja Apple uređaja. Apple koristi napredne alate za usporedbu proizvoda. Mogu se usporediti karakteristike i specifikacije različitih uređaja.

Ikea web trgovina posjeduje detaljne upute za montažu i sklapanje namještaja, a na web trgovini se nalaze videozapisi te korak-po-korak upute. Posjeduje i listu želja, brzu dostavu te mogućnost montaže proizvoda u kući korisnika. Korisnici mogu dizajnirati prostorije svog doma korištenjem aplikacije Ikea koja ima vodiče te alate za mjerenje.

Amazon posjeduje „Prime“ članstvo koje nudi razne pogodnosti kao što su besplatna dostava, brža isporuka te ekskluzivan pristup proizvodima. Korisnici mogu pregledavati recenzije korisnika te ocjene drugih kupaca što im može pomoći u kupnji i odabiru. Amazon web trgovina je dostupna na više različitih lokacija, što ubrzava dostavu te je prevedena na više svjetskih jezika. Amazon koristi vlastitu infrastrukturu odnosno cloud computing AWS (engl. *Amazon Web Services*) računarstvo u oblaku koja im omogućuje otpornost na velika prometna opterećenja.

### **2.3. Sigurnost i privatnost korisnika**

Sve web trgovine koriste osnovne sigurnosne mjere poput SSL (eng. *Secure Sockets Layer*) enkripcije za zaštitu prijenosa podataka, većina pruža dodatne sigurnosne opcije poput dvofaktorske autentikacije kako bi zaštitili korisničke račune. Platforme omogućuju korisnicima upravljanje svojim postavkama privatnosti. Apple i Amazon posebno ističu privatnost korisnika kao ključni element svoje strategije, te zbog toga koriste napredne tehnologije za zaštitu podataka.

### **3. FUNKCIONALNI ZAHTJEVI APLIKACIJE**

Funkcionalni zahtjevi predstavljaju ključni element u razvoju aplikacija jer definiraju što aplikacija treba raditi. Funkcionalni zahtjevi specificiraju osnovne funkcionalnosti i interakcije koje su neophodne za ispunjavanje očekivanja korisnika, te služe kao vodič za razvojni tim tokom cijelog procesa implementacije. Primarni cilj web trgovine je omogućiti korisnicima jednostavno iskustvo kupovine putem interneta, što uključuje pregledavanje proizvoda, dodavanje u košaricu, te sigurno izvršavanje plaćanja. Web trgovina omogućuje korisnicima funkcionalnosti poput dodavanja proizvoda u košaricu, dodavanje i brisanje proizvoda iz košarice, pregled cijene pojedinačnog proizvoda te cijenu svih proizvoda u košarici, integraciju s pružateljima usluga za plaćanje kao što je Stripe koji omogućava sigurne transakcije uz SSL enkripciju, dok sustav automatski obavještava korisnike o statusu njihovih narudžbi putem adrese e-pošte. Također web trgovina posjeduje kontrolnu ploču unutar koje je moguće dodavati proizvode, dodati naziv proizvoda, kategoriju, fotografiju, opis te cijenu proizvoda. Moguće je dodavati razne kategorije proizvoda te njihove potkategorije. Postoji mogućnost brisanja i uređivanja proizvoda unutar kontrolne ploče. Prijava unutar kontrolne ploče moguća je uz pomoć Google računa, te je prijava moguća samo odabranim korisnicima.

## 4. KORIŠTENE TEHNOLOGIJE

Za izradu web aplikacija potrebno je znanje klijentske i poslužiteljske strane programskog razvoja. Na klijentskom (eng. *frontend*) dijelu aplikacije korisnik direktno komunicira s aplikacijom. Glavna svrha frontend razvoja je izrada korisničkog sučelja koje je privlačno, interaktivno, dinamično, prilagođeno za sve uređaje i lagano za korištenje. Osnovne tehnologije te programski jezici koji se koriste u frontend razvoju su HTML (eng. HyperText Markup Language), CSS (eng. Cascading Style Sheet) i JavaScript. HTML služi za strukturu stranice, CSS je zaslužan za stil i dizajn stranice, dok JavaScript dodaje stranici interaktivnost. Postoje i razni razvojni okviri te biblioteke kao što su React.js, Angular.js te Vue.js. To su ujedno i najpopularniji frontend razvojni okviri. Poslužiteljski (eng. *backend*) dio aplikacije obuhvaća sve ono što se događa u pozadini aplikacije, baze podataka i komunikaciju s frontend dijelom aplikacije. Backend je odgovoran za upravljanje podacima te omogućuje da frontend strana pravilno funkcionira. Tehnologije te programski jezici koji se najčešće koriste u backend razvoju su JavaScript te njegovi razvojni okviri Node.js i Express.js, PHP te njegov razvojni okvir Laravel te razvojni okvir Ruby on Rails. Nakon izrade frontend i backend dijela aplikacije, aplikacija se tada naziva full-stack (razvijanje aplikacije na klijentskoj i poslužiteljskoj) aplikacija. Full-stack programeri posjeduju znanja kreiranja web aplikacija na klijentskoj strani i na poslužiteljskoj strani. Prednosti full-stack razvoja su svakako profitabilnost, brži razvoj i rješavanje problema, povećana učinkovitost, fleksibilnost i jednostavnija implementacija.

### 4.1. Next.js

Next.js je popularni razvojni okvir otvorenog koda zasnovan na React.js koji omogućava razvoj klijentskih i poslužiteljskih aplikacija te je u mogućnosti pružiti različite stilove renderiranja i optimizacije performansi. Next.js se koristi za izradu brzih web aplikacija s odličnim SEO (eng. *Search Engine Optimization* – proces poboljšanja kvalitete i kvantitete na web stranici). Next.js koristi SSR (eng. Server-Side Rendering) odnosno omogućava generiranje HTML-a na poslužitelju za svaku stranicu prije nego što se pošalje korisniku. Takav način prikazivanja web aplikacije omogućava brzo učitavanje te poboljšava SEO. Također koristi i API rute unutar aplikacije za frontend. Takve rute omogućavaju komunikaciju s backend stranom bez potrebe za zasebnim backend serverom. API rute se definiraju unutar '/pages/api' direktorija. Next.js dijeli kod u manje dijelove kako bi se optimiziralo učitavanje stranica. To znači da Next.js učitava samo one dijelove koda koji su potrebni za trenutnu

stranicu odnosno za stranicu na kojoj se korisnik trenutno nalazi, čime smanjuje vrijeme učitavanja.

## 4.2. React.js

React.js je popularna biblioteka za izradu korisničkih sučelja odnosno klijentskog dijela aplikacije, koju je razvila kompanija Facebook. React.js omogućava stvaranje složenih, interaktivnih korisničkih sučelja s visokim performansama. React.js koristi komponente koje su samostalni dijelovi koda koji se mogu ponovno koristiti u bilo kojem dijelu aplikacije. One definiraju kako se korisničko sučelje treba prikazati i ponašati. React.js koristi virtualni DOM (eng. Document Object Model). Prilikom svake izmjene na web aplikaciji korisničko sučelje se ponovno prikazuje. React.js koristi JavaScript XML (engl. *eXtensible Markup Language*) odnosno JSX (engl. *JavaScript XML*), a to je proširenje koje omogućuje pisanje HTML-a unutar JavaScripta. JSX kod je pregledniji, olakšava i pojednostavljuje pisanje React.js komponenti jer se struktura i logika nalaze na istom mjestu.

## 4.3. MongoDB

MongoDB je popularna NoSQL (eng. *Structured Query Language* – strukturni upitni jezik) baza podataka koja pohranjuje podatke u formatu sličnom JSON (eng. JavaScript Object Notation – otvoreni format datoteke i format za razmjenu datoteka), poznatom kao BSON (Binary JSON). MongoDB baza podataka je idealan izbor za aplikacije koje zahtijevaju brze promjene strukture podataka, a pruža visoke performanse uz pomoć indeksiranja, brzog pretraživanja i optimiziranog pohranjivanja podataka.

## 4.4. Tailwind CSS

Tailwind CSS je moderni CSS razvojni okvir koji omogućava brzu i efikasnu izradu stilova za web aplikacije. Za razliku od tradicionalnih CSS razvojnih okvira koji dolaze s unaprijed definiranim komponentama, Tailwind pruža niz malih CSS klasa koje se koriste direktno unutar HTML-a, što smanjuje pisanje CSS-a pa nema potrebe za posebnim CSS datotekama, te omogućuje bržu prilagodbu sučelja.

## 5. POSTUPAK IZRADE APLIKACIJE

Za izradu web aplikacije uz pomoć biblioteke Next.js potrebno je instalirati radnu okolinu. Za instalaciju Next.js aplikacije potreban je Node.js i npm (engl. *Node Package Manager*). Potrebno je napraviti novi direktorij pod nazivom *print-alfa*. Direktorij *print-alfa* sadržava direktorije *admin* koji služi za backend, odnosno poslužiteljski dio aplikacije, te „front“ koji služi za frontend, odnosno klijentski dio aplikacije. Također je potreban softver za pisanje koda, u ovom slučaju je to Visual Studio Code.

### 5.1. Opis izrade poslužiteljskih funkcionalnosti

Za instalaciju Next.js potrebno je otvoriti Visual Studio Code te otvoriti terminal u kojem je potrebno upisati sljedeće naredbe:

```
$ npx create-next-app@latest admin
```

Slika 6 - Instalacija Next.js aplikacije

Ovom naredbom u terminalu je napravljena aplikacija za Next.js. Tijekom postavljanja aplikacije, može se odabrati opciju gdje će se Tailwind CSS instalirati automatski. U ovom slučaju Tailwind CSS je ručno instaliran. Nakon toga slijedi instalacija Tailwind CSS razvojnog okvira:

```
npm install -D tailwindcss  
npx tailwindcss init
```

Slika 7 - Instalacija Tailwind CSS razvojnog okvira

U trećem koraku je potrebno otvoriti datoteku *tailwind.config.js* te kopirati sljedeći dio koda:

```
module.exports = {  
  content: [  
    "./app/**/*.{js,ts,jsx,tsx,mdx}",  
    "./pages/**/*.{js,ts,jsx,tsx,mdx}",  
    "./components/**/*.{js,ts,jsx,tsx,mdx}",  
  
    // Or if using `src` directory:  
    "./src/**/*.{js,ts,jsx,tsx,mdx}",  
  ],  
  theme: {  
    extend: {},  
  },  
  plugins: [],  
}
```

Slika 8 - Rute do datoteke



Nakon ovih koraka potrebno je otvoriti datoteku *globals.css* koja se nalazi u direktoriju *styles* (*styles/globals.css*), te obrisati sve i kopirati sljedeći dio koda:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Slika 9 - Tailwind CSS direktive

Nakon instalacije potrebno je pokrenuti server:

```
npm run dev
```

Slika 10 - Pokretanje servera

### 5.1.1. Povezivanje s MongoDB Atlas

Sljedeći korak je povezivanje s bazom podataka. Potrebno je povezati se s MongoDB Atlas. To je usluga baze podataka koja se nalazi u oblaku, a pruža sigurnost i pouzdanost podataka. Potrebno je napraviti račun na MongoDB Atlas platformi, a nakon izrade računa potrebno je napraviti datoteku u direktoriju *admin* pod nazivom *.env*. Unutar datoteke potrebno je dodijeliti vrijednost varijabli *MONGODB\_URI*, kao što je prikazano na slici x:

```
MONGODB_URI = 'mongodb+srv://<username>:<password>@cluster0.mongodb.net/  
myFirstDatabase?retryWrites=true&w=majority|'
```

Slika 11 - Povezivanje s bazom podataka

Potrebno je zamijeniti *<username>* i *<password>* s korisničkim podacima koji odgovaraju bazi podataka. Kod omogućuje povezivanje aplikacije s MongoDB bazom podataka korištenjem Mongoose. Funkcija *mongooseConnect()* provjerava je li veza uspostavljena provjerom izraza (*'readyState === 1'*), a ako je, vraća postojeću vezu. U suprotnom, koristi URI baze podataka iz *.env* datoteke „*process.env.MONGODB\_URI* za uspostavljanje nove veze:

```
export function mongooseConnect() {
  if (mongoose.connection.readyState === 1) {
    return mongoose.connection.asPromise();
  } else {
    const uri = process.env.MONGODB_URI;
    return mongoose.connect(uri);
  }
}
```

Slika 12 - Provjera veze

### 5.1.2. Autentikacija s kontrolnom pločom

Autentikacija za kontrolnu ploču implementirana je korištenjem NextAuth, fleksibilnog i jednostavnog rješenja za autentikaciju u Next.js aplikacijama. NextAuth omogućuje brzo postavljanje prijave i upravljanje sjednicama korisnika uz podršku za različite pružatelje autentikacija, uključujući prijavu putem e-pošte, Google, GitHub, Facebook i mnogih drugih. Autentikacija u kontekstu kontrolne ploče je izrazito važna iz razloga što bi samo ovlašteni korisnici trebali imati pristup osjetljivim funkcijama poput dodavanja, uređivanja ili brisanja proizvoda, kategorija te svih ostalih osjetljivih i zaštićenih podataka i dijelova aplikacije. Korištenjem NextAuth postavljen je sustav prijave putem Google računa. NextAuth upravlja sjednicama korisnika na siguran način, korištenjem enkripcije za pohranu podataka, čime se dodatno povećava sigurnost aplikacije.

### 5.1.3. API ruta za upravljanje proizvodima

U sklopu administracijskog dijela web trgovine, implementiran je API za upravljanje proizvodima korištenjem Next.js i MongoDB. Ovaj API daje dozvolu administratorima za obavljanje ključnih CRUD funkcionalnosti. CRUD funkcionalnosti su raspoređene kroz četiri HTTP (eng. HyperText Transfer Protocol) metode: GET dohvaća sve ili pojedinačne proizvode prema ID-u, POST omogućuje kreiranje novih proizvoda, PUT ažurira postojeće proizvode prema njihovom ID-u, dok DELETE briše proizvode. Funkcija *mongooseConnect()* upravlja vezom s MongoDB bazom podataka, dok *isAdminRequest(req, res)* osigurava da samo ovlašteni korisnici imaju pristup ovim funkcionalnostima. Ovakav pristup jamči sigurnost i učinkovitost u upravljanju podacima o proizvodima unutar sustava.

```

export default async function handle(req, res) {
  const {method} = req;
  await mongooseConnect();
  await isAdminRequest(req,res);
  if (method === 'GET') {
    if (req.query?.id) {
      res.json(await Product.findOne({_id:req.query.id}));
    } else {
      res.json(await Product.find());
    }
  }

  if (method === 'POST') {
    const {title,description,price,images,category,properties} = req.body;
    const productDoc = await Product.create({
      title,description,price,images,category,properties,
    })
    res.json(productDoc);
  }

  if (method === 'PUT') {
    const {title,description,price,images,category,properties,_id} = req.body;
    await Product.updateOne({_id}, {title,description,price,images,category,properties});
    res.json(true);
  }

  if (method === 'DELETE') {
    if (req.query?.id) {
      await Product.deleteOne({_id:req.query?.id});
      res.json(true);
    }
  }
}

```

Slika 13 - API rute za upravljanje

#### 5.1.4. Model za upravljanje podacima u bazi podataka

Modeli definišu strukturu baze podataka te način na koji se podaci pohranjuju i upravljaju u bazi podataka. Za upravljanje podacima o proizvodima unutar web trgovine korišten je *Mongoose* model pod imenom *Product*. Model je definiran pomoću *Mongoose* biblioteke i opisuje strukturu podataka koji se pohranjuje u MongoDB bazu podataka.

```

const ProductSchema = new Schema({
  title: {type:String, required:true},
  description: String,
  price: {type: Number, required: true},
  images: [{type:String}],
  category: {type:mongoose.Types.ObjectId, ref:'Category'},
  properties: {type:Object},
}, {
  timestamps: true,
});

export const Product = models.Product || model('Product', ProductSchema);

```

Slika 14 - Model za spremanje podataka u bazu

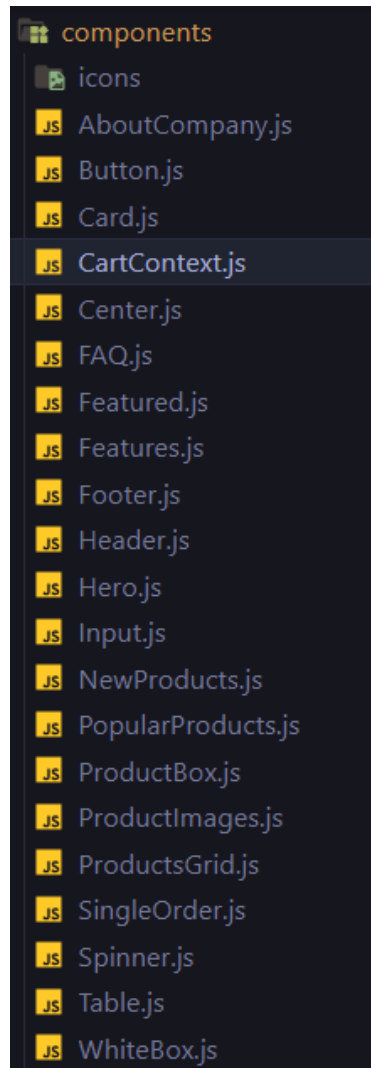
Prednosti modela su u tome što osiguravaju konzistentnost podataka i jednostavan pristup putem API ruta. Također omogućuju validaciju podataka i provjeru određenih kriterija, poput obaveznih polja, te definiranje odnosa između proizvoda i njihovih kategorija u bazi podataka.

## 5.2. Opis izrade klijentske strane aplikacije

Klijentska strana aplikacije odnosi se na razvoj korisničkog sučelja uz korištenje React-a i Next.js-a. Fokus je na izradi modularnih i ponovno iskoristivih komponenti koje omogućuju lakšu organizaciju i održavanje koda. Među ključnim komponentama je i komponenta za navigaciju, koja korisnicima omogućuje jednostavno kretanje kroz različite dijelove aplikacije. Direktorij *pages* u Next.js-u automatski generira rute, olakšavajući upravljanje stranicama i omogućavajući brži razvoj aplikacije. React hooks, poput *useState* i *useEffect*, koriste se za upravljanje stanjem i izvršavanje poslovne logike unutar komponenti, kao što je dodavanje proizvoda u košaricu ili njihovo brisanje. Osim toga, opisuje se kako implementirati Stripe integraciju za sigurno procesuiranje plaćanja, pružajući korisnicima jednostavan i siguran način za obavljanje transakcija.

### 5.2.1. Izrada komponenti

Prvi korak u izradi frontend dijela aplikacije je izrada mape s komponentama koje čine korisničko sučelje. Komponente su blokovi u React.js od kojih se sastoji svaka aplikacija napravljena korištenjem biblioteke React.js. Omogućuje ponovnu upotrebu koda, održivost aplikacije i modularnost. Izrada komponenti je princip podjele sučelja na manje dijelove koji se mogu nezavisno razvijati i testirati.



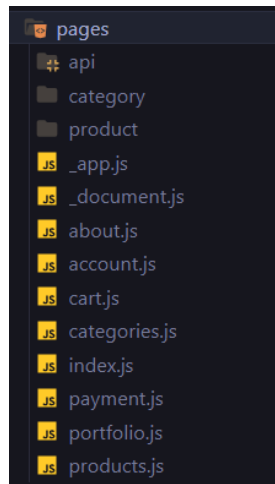
Slika 15 - Popis komponenti

### 5.2.2. Komponenta za navigaciju

Navigacija je važna komponenta svake web aplikacije jer korisnicima omogućava jednostavno kretanje po web aplikaciji. Za uspješno kretanje po web aplikaciji potrebno je koristiti *Link* komponentu iz Next.js. *Link* komponenta se koristi za navigaciju između stranica. Pruža optimiziranu klijentsku navigaciju te omogućava stvaranje dinamičkih ruta.

### 5.2.3. Direktorij pages

Kako bi navigacijska komponenta uspješno radila, u direktoriju *pages* potrebno je napraviti podstranice unutar direktorija *pages*, u ovom slučaju su to stranice *products.js* i *portfolio.js*. Uz pomoć direktorija *pages* moguće je jednostavno kretanje po web aplikaciji, svaka stranica je neovisna komponenta, što olakšava održavanje i razvoj.



Slika 16 - Direktorij pages

#### 5.2.4. React Hooks

React se konstantno proširuje te dobiva nove značajke i komponente. React posjeduje takozvane kuke (eng. *Hooks*), odnosno to su značajke koje nam omogućuju korištenje različitih stanja, ali i ostalih značajki Reacta bez potrebe pisanja klase. Hookovi posjeduju bitna pravila, pozivaju se samo na najvišoj razini, nije preporučeno pozivati ih unutar petlji, ugniježđenih funkcija ili uvjeta. Dvije najpoznatije i najkorištenije kuke su *useState* i *useEffect*. Funkcionalnost *useState* komponente nam omogućuje postavljanje i kontroliranje *state* u funkcijskim komponentama. *State* čine svi podaci u komponenti koji će se možda mijenjati vremenom. *UseEffect* je jedan od osnovnih kuka, omogućuje izvršavanje sporednih efekata unutar funkcionalnih komponenti, neki od sporednih efekata su dohvaćanje podataka iz API-a, manipulaciju DOM-om, postavljanje mjerača vremena (eng. *Timer*). *UseEffect* omogućuje jednostavno dodavanje i upravljanje efektima unutar funkcionalnih komponenti. Korištenjem *useEffect* kuke, React omogućuje upravljanje sporednim efektima na deklarativan način, što olakšava razumijevanje i održavanje funkcionalnih komponenti.

```

useEffect(() => {
  if (cartProducts.length > 0) {
    axios.post('/api/cart', {ids:cartProducts})
      .then(response => {
        setProducts(response.data);
      })
  } else {
    setProducts([]);
  }
}, [cartProducts]);
useEffect(() => {
  if (typeof window === 'undefined') {
    return;
  }
  if (window?.location.href.includes('success')) {
    setIsSuccess(true);
    clearCart();
  }
}, []);

```

Slika 17 - useEffect

### 5.2.5 Logika dodavanja i brisanja proizvoda unutar košarice

Kako bi košarica ispravno radila potrebno je napraviti logiku za dodavanje proizvoda u košaricu klikom na gumb *Dodaj u košaricu*. Upravljanje košaricom je važno jer korisnici mogu dodavati, brisati te pregledavati proizvode unutar košarice prije završetka kupovine. Kako bi implementacija logike bila uspješna korištena je komponenta „CartContext.js“. Vrlo važan dio koda unutar komponente su funkcije *createContext* i *useContext*. Funkcija *createContext* se koristi za stvaranje konteksta, sadrži dvije glavne komponente koje se zovu *Provider* i *Consumer*. *Provider* komponenta omogućuje pristup podacima komponentama koje su unutar nje, dok *Consumer* komponenta omogućava pristup vrijednostima iz konteksta.

```

export function CartContextProvider({children}) {
  const ls = typeof window !== "undefined" ? window.localStorage : null;
  const [cartProducts, setCartProducts] = useState([]);
  useEffect(() => {
    if (cartProducts?.length > 0) {
      ls?.setItem('cart', JSON.stringify(cartProducts));
    }
  }, [cartProducts]);
  useEffect(() => {
    if (ls && ls.getItem('cart')) {
      setCartProducts(JSON.parse(ls.getItem('cart')));
    }
  }, []);
  function addProduct(productId) {
    setCartProducts(prev => [...prev, productId]);
  }
  function removeProduct(productId) {
    setCartProducts(prev => {
      const pos = prev.indexOf(productId);
      if (pos !== -1) {
        return prev.filter((value, index) => index !== pos);
      }
      return prev;
    });
  }
  function clearCart() {
    setCartProducts([]);
  }
  return (
    <CartContext.Provider value={{cartProducts, setCartProducts, addProduct,
    removeProduct, clearCart}}>
      {children}
    </CartContext.Provider>
  );
}

```

Slika 18 - Funkcija za dodavanje proizvoda u košaricu

## 5.2.6 Stripe integracija

Unutar web trgovine nalazi se i integracija s web servisom za naplatu Stripe. Stripe je popularna platforma za procesiranje plaćanja koji omogućuje jednostavno i sigurno upravljanje transakcijama. Stripe je odabran zbog široke podrške za različite načine plaćanja te zbog jednostavne integracije u web trgovinu. Stripe omogućava jednostavno plaćanje putem web trgovine, omogućava razne načine plaćanja poput bankovnih kartica, digitalnih novčanika. Kako bi korisnici izvršili plaćanje potrebno je unijeti podatke s kartice te potvrditi narudžbu. Stripe API integracija omogućuje lako praćenje transakcija te povrate novca.



## 6. PRIKAZ RADA APLIKACIJE

Nakon pokretanja aplikacije, korisnik dolazi na početnu stranicu gdje može pregledavati proizvode prema kategorijama. Korištenjem navigacije, omogućene preko Next.js-a, korisnik može lako prelaziti između različitih dijelova web shopa, poput stranice s kategorijama, detaljima proizvoda ili košaricom. Kada korisnik odluči dodati proizvod u košaricu, taj se proizvod sprema putem CartContext komponente, omogućavajući da podaci budu dostupni na svim stranicama. Nakon što je košarica popunjena, korisnik može pristupiti stranici za plaćanje, gdje se integracija sa Stripe servisom koristi za sigurno obavljanje transakcija.

### 6.1. Prikaz rada aplikacije – klijentska strana

U ovom poglavlju se nalaze uputstva za korištenje web trgovine, korištenje web trgovine vrlo je jednostavno. Web trgovina nema opciju prijave i registracije. U gornjem desnom kutu se nalazi gumb s ikonom košarice, klikom na taj gumb navigacija odvodi na stranicu košarice gdje se nalaze svi proizvodi koji su u košarici ili se otvori prazna stranica u kojoj piše da je košarica prazna. U lijevom kutu se nalazi logo tvrtke koji je ujedno i gumb, klikom na taj gumb navigacija vraća na početnu stranicu. U navigaciji se nalaze još dva gumba, jedan je pod imenom *Proizvodi*, klikom na taj gumb navigacija vodi na stranicu gdje se nalaze svi proizvodi koji su dostupni, drugi gumb je pod imenom *Portfolio*, klikom na taj gumb navigacija vodi na stranicu gdje se nalazi mapa radova. Na početnoj stranici se također nalaze dva gumba *Proizvodi* i *Naš portfolio*.



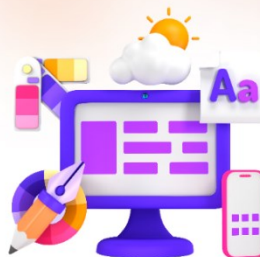
Slika 19 - Početna stranica "Print Alfa"

Izvor: Autor

Print Alfa

## Ekspteri za grafički dizajn, web razvoj i programiranje

Print Alfa, osnovana s vizijom inovacije u dizajnu i preciznosti u tisku, pruža izvanredne usluge u području grafičkog dizajna, tiska i web razvoja. Naša misija je transformirati vaše poslovne ideje u vizualno impresivne proizvode koji komuniciraju i inspiriraju.



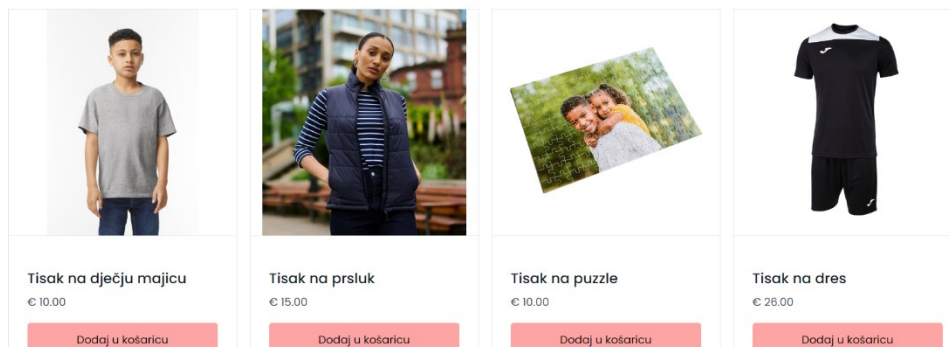
Slika 20 - Sekcija o opisu tvrtke

Izvor: Autor

U ovom dijelu stranice su navedeni novi proizvodi koji su u ponudi, klikom na gumb Dodaj u košaricu, proizvod će se dodati u košaricu. U košaricu se može dodati jedan ili više proizvoda, također moguće je kliknuti na proizvod te će se otvoriti stranica s proizvodom te svim informacijama o proizvodu.

### Proizvodi

Otkrijte našu raznoliku ponudu proizvoda, dizajniranih da zadovolje sve vaše potrebe. Od najnovijih inovacija do bezvremenskih klasika, naša kolekcija nudi nešto za svakoga. Pregledajte i pronadite savršen izbor za vaš stil života, ukus i budžet.



Slika 21 - Novi proizvodi na web trgovini

Izvor: Autor

Nakon klika na određeni proizvod, stranica vodi na zasebnu stranicu određenog proizvoda, u ovom slučaju je to stranica proizvoda pod nazivom Tisak na dječju majicu. Ovdje

se nalazi opis proizvoda, cijena proizvoda, fotografije proizvoda te gumb *Dodaj u košaricu*, putem kojeg je također moguće dodati proizvod u košaricu.



### Tisak na dječju majicu

Naša usluga personaliziranog tiska na dječje majice omogućuje vam da stvorite jedinstvene i zabavne majice koje će djeca obožavati nositi. Izradene od mekih i udobnih materijala, naše dječje majice su savršene za svakodnevno nošenje, a s visokokvalitetnim tiskom, omogućuju vam da istaknete bilo koji dizajn, logotip ili poruku. Bez obzira koristite li ih kao poklon za rođendane ili posebne prigode, kao dio školskih ili sportskih događanja, ili čak za promociju dječjih događanja ili proizvoda, naše personalizirane dječje majice su izvrstan način da se istaknete. S raznolikim opcijama dizajna, uključujući različite boje, ilustracije i fontove, možete stvoriti majice koje odražavaju individualnost svakoga djeteta. Udobne, atraktivne i

€ 10.00

★★★★★ 117 recenzija

Dodaj u košaricu

Slika 22 - Stranica proizvoda



Izvor: Autor

Ovo je stranica košarice, ovdje se nalaze svi proizvodi koji su dodani u košaricu, ovdje je moguće pregledati cijene proizvoda te ukupnu cijenu košarice. Klikom na + u košaricu se dodaje još jedan proizvod, klikom na – proizvod se uklanja iz košarice ako je njegova vrijednost 1, ako je vrijednost proizvoda veća od 1 onda se broj proizvoda umanjuje za 1.



Proizvodi Portfolio

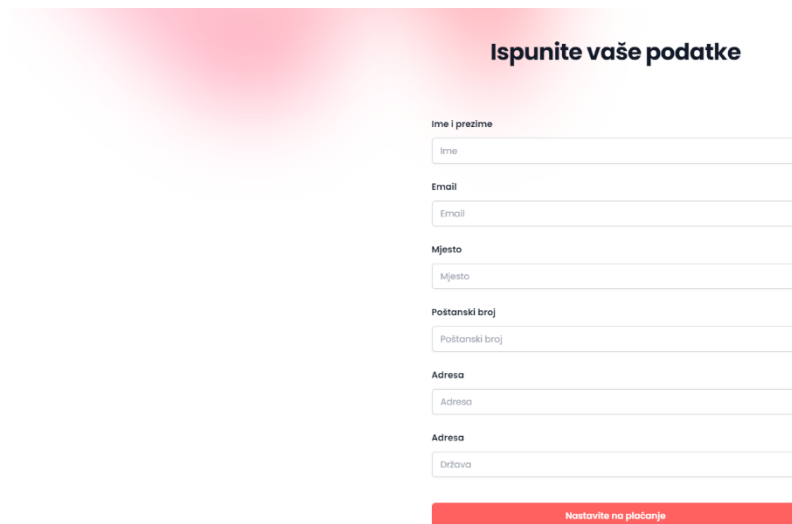


Vaša košarica	KOLIČINA	CIJENA
 Tisak na prsluk	- 1 +	€ 15
 Tisak na dječju majicu	- 1 +	€ 10
		€ 25
<a href="#">Sjedeći korak</a>		

Slika 23 - Stranica košarice

Izvor: Autor

Unutar stranice gdje se nalazi košarica se nalazi i gumb *Sljedeći korak*, klikom na taj gumb otvara se nova stranica u koju je potrebno ispuniti formu s imenom i prezimenom, emailom, mjestom stanovanja, poštanskim brojem te adresom.



Ispunite vaše podatke

Ime i prezime

Email

Mjesto

Poštanski broj

Adresa

Adresa

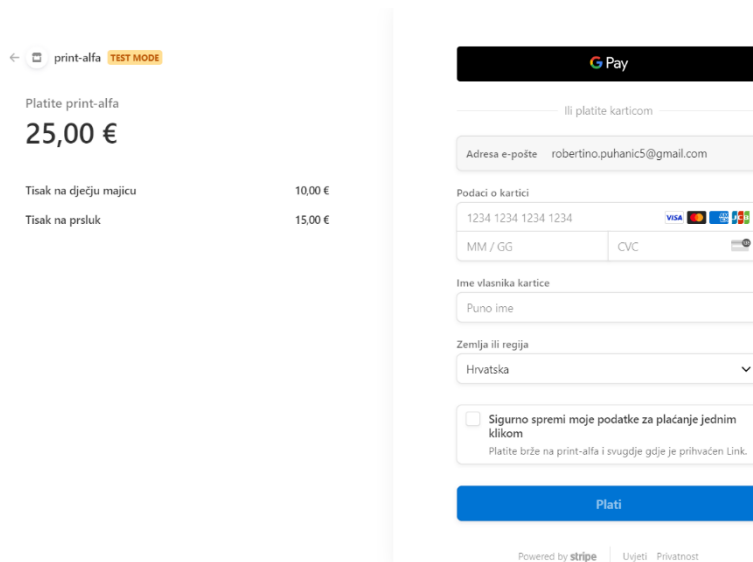
Država

Nastavite na plaćanje

Slika 24 - Stranica za ispunjavanje podataka

Izvor: Autor

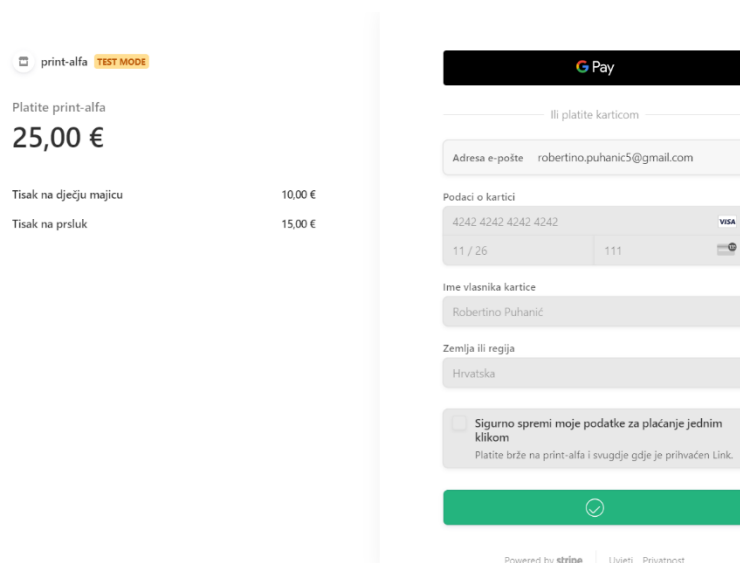
Nakon što se podaci ispune nalazi se gumb *Nastavite na plaćanje*, nakon klika na gumb, prikazana je nova stranica, odnosno na stranica za plaćanje. Stranica za plaćanje je napravljena s Stripe integracijom. Sve transakcije se obavljaju uz pomoć Stripe servisa. Na ovoj stranici su dostupne informacije poput ukupnog iznosa koji je potrebno platiti, ovdje se nalazi i popis proizvoda koje naručujemo iz web trgovine te cijena svakog proizvoda. Plaćanje je moguće izvršiti uz pomoć Google Pay-a ili uz pomoć bankovne kartice. Potrebno je ispuniti formu, unijeti podatke o kartici odnosno broj kartice, datum isteka te sigurnosni broj, ime vlasnika kartice te zemlju, u ovom slučaju je to Hrvatska. Moguće je odabrati opciju gdje će se podaci spremići te moći ponovno koristiti prilikom druge kupovine. Nakon što se svi podaci uspješno ispune potrebno je kliknuti na gumb *Plati*.



Slika 25 - Obrazac za upis kartice i izvršavanje plaćanja

Izvor: Autor

Nakon klika na gumb *Plati* plavi gumb mijenja boju u zeleni te tako se dobiva informacija da je plaćanje uspješno izvršeno.



Slika 26 - Uspješno izvršeno plaćanje

Izvor: Autor

## 7. ZAKLJUČAK

Izrada moderne web trgovine korištenjem Next.js, React.js, MongoDB i Tailwind CSS pokazuje kako napredne tehnologije mogu unaprijediti proces razvoja web aplikacija, pružajući korisnicima besprijekorno iskustvo online kupovine. Aplikacija je uspješno izrađena s jasnom podjelom na poslužiteljski i klijentski dio, čime je postignuta visoka razina performansi i sigurnosti. Poslužiteljski dio omogućuje stabilno i sigurno upravljanje podacima, dok klijentski dio pruža responzivno korisničko sučelje.

Integracijom vanjskih servisa kao što su NextAuth i Stripe, aplikacija zadovoljava visoke standarde sigurnosti i pouzdanosti, pružajući korisnicima povjerenje u sigurno obavljanje transakcija i zaštitu njihovih podataka. Modularna struktura koda, s naglaskom na ponovno korištenje komponenti i upravljanje stanjem aplikacije putem React kuka, olakšava buduće održavanje i nadogradnju aplikacije. Ovaj projekt ne samo da ispunjava trenutne potrebe tržišta za modernim rješenjima u online kupovini, već nudi i fleksibilnu platformu koja je spremna za buduće izazove i zahtjeve korisnika.

Aplikacija bi se mogla unaprijediti dodavanjem mogućnosti za sortiranje proizvoda prema cijeni, imenu, datumu objave, kao i kategorizacijom proizvoda. Također bi se mogla omogućiti prijava korisnika na klijentskoj strani i označavanje omiljenih proizvoda. Unutar korisničkog profila bi se pohranjivale sve narudžbe i omiljeni proizvodi.

## LITERATURA

1. React URL: <https://react.dev/learn> [pristup: 01.09.2024.]
2. Next.js URL: <https://nextjs.org/docs> [pristup: 01.09.2024.]
3. Tailwind CSS URL: <https://tailwindcss.com/docs/installation> [pristup: 01.09.2024.]
4. MongoDB URL: <https://www.mongodb.com/docs/> [pristup: 01.09.2024.]
5. A. Fedosejev (2015), *React.js Essentials*, Packt
6. React Hooks URL: <https://react.dev/reference/react/hooks> [pristup: 28.08.2024.]
7. Create Next App URL: <https://nextjs.org/docs/getting-started/installation> [pristup: 03.09.2024]

## POPIS SLIKA

Slika 1 - Zalando web trgovina .....	2
Slika 2 - Ebay web trgovina .....	3
Slika 3 - Apple web trgovina.....	4
Slika 4 - Ikea web trgovina.....	4
Slika 5 - Amazon web trgovina.....	5
Slika 6 - Instalacija Next.js aplikacije .....	10
Slika 7 - Instalacija Tailwind CSS razvojnog okvira .....	10
Slika 8 - Rute do datoteke .....	10
Slika 9 - Tailwind CSS direktive.....	11
Slika 10 - Pokretanje servera.....	11
Slika 11 - Povezivanje s bazom podataka .....	11
Slika 12 - Provjera veze .....	11
Slika 13 - API rute za upravljanje .....	13
Slika 14 - Model za spremanje podataka u bazu .....	13
Slika 15 - Popis komponenti .....	15
Slika 16 - Direktorij pages .....	16
Slika 17 - useEffect .....	17
Slika 18 - Funkcija za dodavanje proizvoda u košaricu.....	18
Slika 19 - Početna stranica "Print Alfa" .....	19
Slika 20 - Sekcija o opisu tvrtke.....	20
Slika 21 - Novi proizvodi na web trgovini .....	20
Slika 22 - Stranica proizvoda .....	21
Slika 23 - Stranica košarice .....	21
Slika 24 - Stranica za ispunjavanje podataka .....	22
Slika 25 - Obrazac za upis kartice i izvršavanje plaćanja .....	23
Slika 26 - Uspješno izvršeno plaćanje.....	23



**IZJAVA O AUTORSTVU RADA**

Ja, **Robertino Puhanić**, pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog/diplomskog rada pod naslovom: **Moderan pristup izrade web aplikacije** te da u navedenom radu nisu na nedozvoljen način korišteni dijelovi tuđih radova.

U Požegi, 11. rujan 2024.

Potpis studenta

