

Izrada web aplikacije za prodaju mobilnih uređaja

Vinski, Ivan

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:794991>

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

Download date / Datum preuzimanja: **2024-07-17**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski studij informatike

Ivan Vinski

Izrada web aplikacije za prodaju mobilnih uređaja

Završni rad

Mentor: doc. dr. sc. Lucia Načinović Prskalo

Rijeka, 10.9.2021.

Rijeka, 2.5.2021.

Zadatak za završni rad

Pristupnik: Ivan Vinski

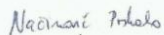
Naziv završnog rada: Izrada web aplikacije za prodaju mobilnih uređaja

Naziv završnog rada na eng. jeziku: Development of a web application for the sales of mobile devices

Sadržaj zadatka: Glavni zadatak završnog rada je izraditi web aplikaciju za prodaju mobilnih uređaja. Pritom će se koristiti odabrani alati za razvoj frontend i backend dijela aplikacije koji će se u radu i detaljno opisati. Također će se opisati i demonstrirati svi važni elementi i funkcionalnosti izrađene web aplikacije.

Mentor

Doc. dr. sc. Lucia Načinović Prskalo



Voditelj za završne radove

doc. dr. sc. Miran Pobar



Zadatak preuzet: 2.5.2021.



(potpis pristupnika)

Sadržaj

| | |
|---|----|
| Sažetak | 4 |
| 1. Uvod | 5 |
| 2. Opis tehnologija korištenih za izradu web aplikacije | 6 |
| 2.1. Alati i programski jezici za razvoj sučelja | 6 |
| 2.1.1. Visual Studio Code | 6 |
| 2.1.2. JavaScript..... | 7 |
| 2.1.3. HTML..... | 7 |
| 2.1.4. CSS | 8 |
| 2.1.5. Quasar framework | 9 |
| 2.1.6. Vue.js..... | 9 |
| 2.1.7. Node.js | 10 |
| 2.1.8. Firebase | 10 |
| 2.2. Skladište podataka..... | 10 |
| 2.2.1. Cloud Storage | 10 |
| 2.3. Baza podataka | 11 |
| 2.3.1. Cloud Firestore | 12 |
| 2.4. Web server | 13 |
| 2.4.1. 000webhost..... | 13 |
| 3. Opis elemenata i funkcionalnosti web aplikacije..... | 14 |
| 3.1. Main layout | 14 |
| 3.1.1. Header..... | 14 |
| 3.1.2. Page container | 17 |
| 3.1.3. Footer..... | 17 |
| 3.2. Opis stranica..... | 17 |
| 3.2.1. Početna stranica | 17 |
| 3.2.2. Prijava/Registracija | 19 |
| 3.2.3. Prikaz proizvoda..... | 20 |
| 3.2.4. Košarica | 24 |
| 3.2.5. Moj račun | 26 |
| 3.2.6. Lista želja | 27 |
| 3.2.7. Narudžbe | 28 |
| 3.2.8. O nama/Dostava/Plaćanje..... | 30 |
| 3.3. Programski kod..... | 31 |
| 3.3.1. Povezivanje web aplikacije s firebase projektom | 31 |
| 3.3.2. Router | 32 |

| | | |
|--------|------------------------|----|
| 3.3.3. | Prijava..... | 34 |
| 3.3.4. | Registracija | 36 |
| 3.3.5. | Početna stranica | 37 |
| 4. | Zaključak..... | 40 |
| 5. | Literatura | 41 |
| 6. | Popis priloga | 42 |
| 7. | Popis slika | 42 |

Sažetak

Ovaj završni rad u uvodnom dijelu opisuje osnovne pojmove web aplikacija te njihovu svrhu, rast i razvoj u budućnosti. U okviru završnog rada, izrađena je web aplikacija „SmartShop“ koja korisnicima omogućuje pregled, filtriranje, sortiranje, pretraživanje, spremanje u listu želja, dodavanje u košaricu te na kraju kupovinu mobilnih uređaja, pametnih satova i dodataka. Korisnici s dodatnim pravima, tj. administratorima, omogućeno je i dodavanje novih proizvoda, izmjena i brisanje postojećih proizvoda te dodavanje i brisanje slika na početnoj stranici.

U radu su detaljno opisane tehnologije koje su se koristile pri izradi web aplikacije, svi elementi i funkcionalnosti aplikacije, a dodatno su opisani i dizajn i programski kôd aplikacije.

Ključne riječi: web aplikacije, e-commerce aplikacije, razvojni okvir, baza podataka, web server

1. Uvod

„Web aplikacije su programska rješenja kojima se pristupa putem internet preglednika koristeći internet. Nagli rast i razvoj web aplikacije trebaju zahvaliti činjenici da su dostupne u bilo koje vrijeme s bilo kojeg mjesta, i računala i mobilnih telefona.“ (horizont.com.hr, 2021.)

Tema ovog završnog rada je izrada web aplikacije za prodaju mobilnih uređaja stoga ćemo govoriti o e-commerce aplikacijama. E-commerce aplikacije koriste se za kupnju i prodaju robe i usluga putem interneta. Neke od najpoznatijih svjetskih e-commerce platforma su amazon.com, ebay.com, aliexpress.com, itd. U Hrvatskoj od poznatijih imamo ekupi.hr, mall.hr, abrakadabra.com, itd. E-commerce aplikacije koje su prvenstveno okrenute prodaji mobilnih uređaja u Hrvatskoj su mobis.hr, sancta-domenica.hr, aloalo.hr i druge.

Razvojem tehnologije i sve lakšim pristupom internetu, e-commerce postaje sve popularniji, a pandemija koja nas je pogodila uvelike je pogodovala razvoju e-trgovina širom svijeta. Kako su mnoge države uvele lockdown i ograničile poslovanje mnogih sektora, trgovci su bili prisiljeni sve više svoje poslovanje okrenuti prema online prodaji. Potrošači su također, kao i trgovci, svoje navike kupovine okrenuli prema e-trgovinama.

„Usljed pandemije COVID-19 i ograničene slobode kretanja mnogi građani su počeli kupovati online pa se kupovina na internetskim stranicama povećala za 15,5% u odnosu na ožujak 2020. godine, a čak 67% ispitanika naglasilo je da online kupuje češće u odnosu na isto razdoblje prošle godine.“ (Matić, 2020.)

No bez obzira na povećano korištenje e-trgovina od strane potrošača, trgovci koji se ne budu prilagodili novim tehnologijama i stalno unapređivali svoje e-commerce aplikacije, brzo će propasti. Razlog tomu je što su upravo potrošači ti koji se vrlo brzo prilagođavaju novim tehnologijama. Tako, primjerice, ako neka e-commerce aplikacija nije prilagođena za upotrebu preko mobitela, veliki dio potrošača koji kupuju preko mobilnih telefona zaobilazit će takve e-trgovine.

U okviru završnog rada izrađena je aplikacija za prodaju mobilnih uređaja, pametnih satova i dodataka „SmartShop“.

Završni se rad sastoji od uvoda, opisa tehnologija, opisa elemenata i funkcionalnosti web aplikacije te zaključka. U sljedećem poglavlju opisuju se tehnologije koje su se koristile za izradu web aplikacije. Zatim slijedi glavni dio završnog rada u kojem se opisuju svi elementi i način funkcioniranja web aplikacije, njezin dizajn i programski kod.

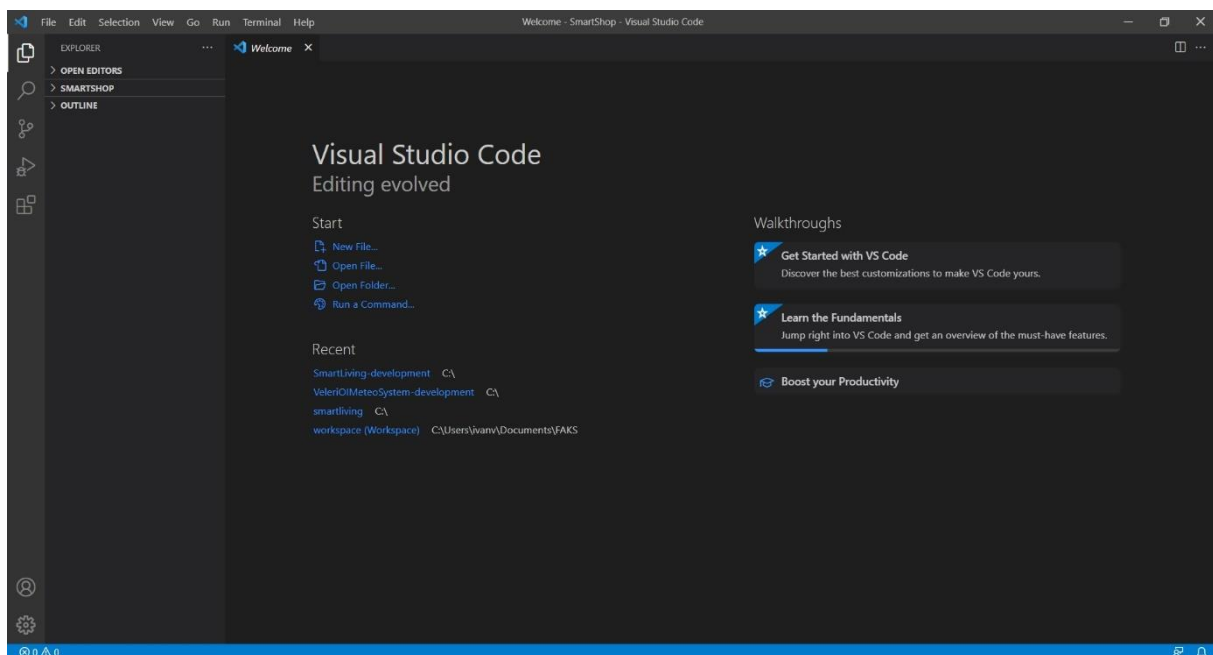
2. Opis tehnologija korištenih za izradu web aplikacije

2.1. Alati i programski jezici za razvoj sučelja

Prije početka izrade bilo kakve web aplikacije potrebno je odabrati alate i tehnologiju pomoću kojih ćemo izraditi cjelokupnu web aplikaciju. Stoga će u ovom poglavlju biti opisani alati i tehnologija koja se koristila za izradu „SmartShop” web aplikacije.

2.1.1. Visual Studio Code

Pisanje programskog koda odvija se unutar softvera odnosno računalnog programa. Programski kod „SmartShop” web aplikacije pisao se pomoću besplatnog računalnog programa Visual Studio Code. Na *Slici 1.* prikazana je početna stranica VS Code-a. To je integrirano razvojno okruženje koje programeru uvelike pomaže prilikom pisanja programskog koda. Pisanje koda moguće je i unutar nekih vrlo jednostavnih programa kao što su Notepad ++, no korištenje programa namijenjenih pisanju programskog koda ima mnogo prednosti. Neke od njih su podrška za ispravljanje pogrešaka, isticanje sintakse, automatsko dovršavanje koda i ostalo. Podrška za ispravljanje pogrešaka pomaže programeru tako što označuje mjesto u kodu na kojem se nalazi sintaksna pogreška te u nekim slučajevima nudi i način rješavanja pogreške. Isticanje sintakse u različitim bojama povećava čitljivost cjelokupnog koda. Automatsko dovršavanje koda uvelike ubrzava proces pisanja programskog koda. Visual Studio Code dolazi sa ugrađenom podrškom za JavaScript, TypeScript i Node.js te bogatom kolekcijom proširenja za ostale programske jezike kao što su C++, C#, Python, PHP, Java i drugi. Jedno od proširenja koje se koristilo prilikom izrade „SmartShop” web aplikacije je Vetur. Vetur je proširenje za isticanje programskog koda koji se koristi uz Vue.js framework.



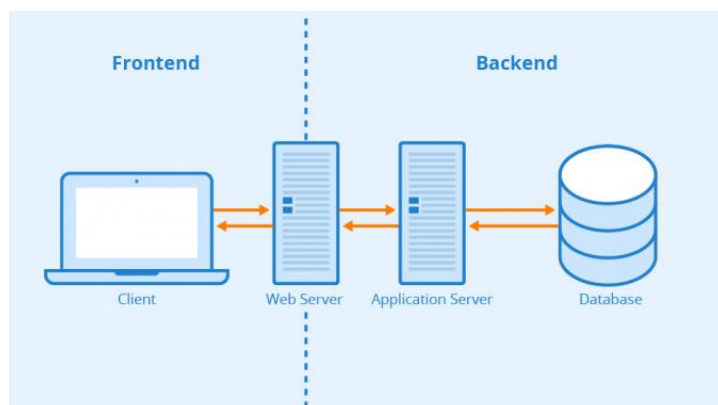
Slika 1. Izgled početne stranice VS Code-a

2.1.2. JavaScript

JavaScript je skriptni programski jezik koji se izvršava u web pregledniku na strani korisnika. JavaScript se prvenstveno koristi za frontend razvoj web aplikacija, no uz pomoć raznih framework-a moguće ga je koristiti i kao backend. *Slika 2.* prikazuje proces razmjene podataka između frontend-a i backend-a. Frontend web razvoj je razvoj grafičkog sučelja web aplikacije na strani korisnika. Odnosno JavaScript, uz HTML i CSS, je zaslužan za sve ono što korisnik vidi na svojem ekranu i upravo on čini web stranice dinamičnima.

„JavaScript je izvrstan alat za zadatke poput provjere valjanosti obrazaca za slanje ili ažuriranje samo određenih dijelova sadržaja stranice bez istodobnog ažuriranja cijele stranice. Bez JavaScripta, Internet bi bio ispunjen web stranicama koje samo miruju i prikazuju tekst.“ (Fowler, 2020.)

Backend razvoj se odnosi na onaj dio programskog koda koji je skriven od korisnika, odnosno korisnik mu ne može pristupiti. Više o backend razvoju biti će opisano u poglavlju 2.1.7. Node.js.



Slika 2. Primjer Frontend vs. Backend - Autor: Seobility - License: [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

2.1.3. HTML

HTML (engl. **H**ypertext **M**arkup **L**anguage) je jezik za označavanje dijelova teksta na webu te definiranje strukture sadržaja. Prilikom izrade „SmartShop“ web aplikacije upotrebljavala se zadnja verzija, odnosno HTML5 koji je standardiziran od strane World Wide Web Consortium-a (W3C).

HTML se sastoji od raznih elemenata uz pomoću kojih oblikujemo strukturu sadržaja da se prikazuje ili ponaša na točno određeni način.

HTML element se sastoji od skupova oznaka (engl. tag) i sadržaja. Sadržaj se nalazi između dvije oznake, početne i završne. Element također može sadržavati i attribute koji se pišu unutar početne oznake. Attribute sadrže dodatne informacije o elementu za koje ne želimo da se prikazuju u stvarnom sadržaju. *Slika 3.* prikazuje primjer jednog HTML elementa.



Slika 3. Primjer HTML elementa

2.1.4. CSS

CSS (engl. **Cascading Style Sheets**) je stilski jezik koji opisuje kako će se prikazivati HTML elementi. Možemo reći da pomoću CSS-a „proljepšavamo“ prikaz sadržaja na web aplikacijama. Neke od mogućnosti CSS-a su promjena veličine fonta, boje teksta, način poravnanja, dodavanje razmaka, raspored elemenata i mnogi drugi.

Postoje tri načina za uključivanje CSS-a unutar dokumenta. Prvi način je pomoću atributa „style“ unutar HTML elementa (Slika 4.). Ovaj način se naziva „Inline CSS“. Drugi način se naziva „Internal CSS“ i on se dodaje unutar `<style>` elementa HTML stranice (Slika 5.). Treći i zadnji način se zove „External CSS“. Ovo je jedan od najboljih načina korištenja CSS-a zbog toga što lako možemo mijenjati izgled cijele web aplikacije promjenom samo jedne datoteke. External CSS se nalazi unutar posebne datoteke koju je potrebno uključiti u HTML dokument pomoću `<link>` elementa (Slika 6.).

```

<!DOCTYPE html>
<html>
  <body>
    <h1 style="color:blue;text-align:center;">Ovo je naslov</h1>
    <p style="color:red;">Ovo je odlomak.</p>
  </body>
</html>

```

Slika 4. Primjer Inline CSS-a

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: linen;
      }

      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>

```

Slika 5. Primjer Internal CSS-a

```

<link rel="stylesheet" href="stil.css">

```

Slika 6. Način uključivanja External CSS-a

2.1.5. Quasar framework

„Framework je platforma za razvoj softverskih aplikacija. Pruža temelje na kojima programeri softvera mogu graditi programe za određenu platformu.“ (Christensson, 2013.)

Quasar je razvojni okvir (engl. framework) baziran na Vue.js razvojnom okviru koji omogućuje web programerima brzu izradu responzivnih web aplikacija. Služi za razvoj prednjeg dijela web aplikacija, odnosno frontend-a. Quasar sadrži gotove programske kodove koje programer uključuje i koristi prilikom izrade web aplikacije. To uveliko ubrzava razvoj web aplikacije jer programer umjesto da mora sve funkcionalnosti pisati sam, samo ih dodaje u kod po potrebi. Neke od funkcionalnosti koje Quasar pruža su padajući izbornik, prikaz dijaloga i notifikacija, gotove liste i tablice, paginacija, steper, tabovi itd.

2.1.6. Vue.js

„Vue.js je progresivni JavaScript okvir (engl. lframework) otvorenog koda za izgradnju korisničkih sučelja (engl. user interface, UI) i single-page aplikacija. Ovaj okvir koristi „visoko razdvajanje“, omogućavajući programerima da postupno stvaraju korisnička sučelja.“ (Mamani, 2019.)

Vue.js je jednostavan za korištenje, lako se uči pa je iz tih razloga prikladan za početnike. Jedna vrlo korisna komponenta Vue.js-a je Vue Router. Routing omogućuje korisniku kretanje između stranica bez potrebe za osvježavanjem stranica. Na taj način smo napravili navigaciju web aplikacije vrlo jednostavnom.

2.1.7. Node.js

„Node.js je razvojna platforma otvorenog koda za izvršavanje JavaScript koda na strani poslužitelja, odnosno servera. Node.js je koristan za razvoj aplikacija koje zahtijevaju trajnu vezu preglednika sa poslužiteljem i često se koristi za aplikacije u stvarnom vremenu, kao što su chat, sažeci vijesti i web push obavijesti.“ (Denman, 2012.)

Node.js nam omogućuje korištenje JavaScript-a za backend razvoj web aplikacija.

Za razliku od frontend-a koji se izvršava na strani korisnika u web pregledniku, backend se izvršava na strani poslužitelja, odnosno servera. Backend kod zaprima zahtjeve od korisnika te sadrži logiku za slanje odgovarajućih podataka nazad klijentu. Backend također uključuje i bazu podataka koja će trajno pohranjivati sve podatke vezane za aplikaciju.

2.1.8. Firebase

Firebase je platforma za kreiranje mobilnih i web aplikacija koju je razvio Google. Sastoji se od različitih servisa kao što su analitika, autentifikacija, baza podataka, skladište podataka, hosting i mnogi drugi. Za potrebe „SmartShop“ web aplikacije koristila su se tri servisa, a to su Firebase Authentication, Cloud Firestore i Cloud Storage.

Firebase nam nudi dva cjenovna plana, Spark Plan i Blaze Plan. Spark Plan je potpuno besplatan do određene količine podataka i sa ograničenim mogućnostima. Baza podataka je ograničena na 1 GB, a Firebase Storage na 5 GB. Blaze Plan je osmišljen na način da plaćamo onoliko koliko potrošimo. U Blaze Plan je također uključen i besplatan Spark Plan, a tek nakon što prekoračimo besplatne mogućnosti počinje se naplaćivati dodatna memorija koja je utrošena. Za „SmartShop“ web aplikaciju dovoljan je besplatni Spark Plan.

2.2. Skladište podataka

Za razliku od baze podataka u kojoj se spremaju informacije, skladište podataka nam služi za spremanje većih datoteka, kao što su fotografije, videozapisi i audio datoteke. „SmartShop“ web aplikacija koristi Cloud Storage za pohranu fotografija koje su generirane od strane korisnika.

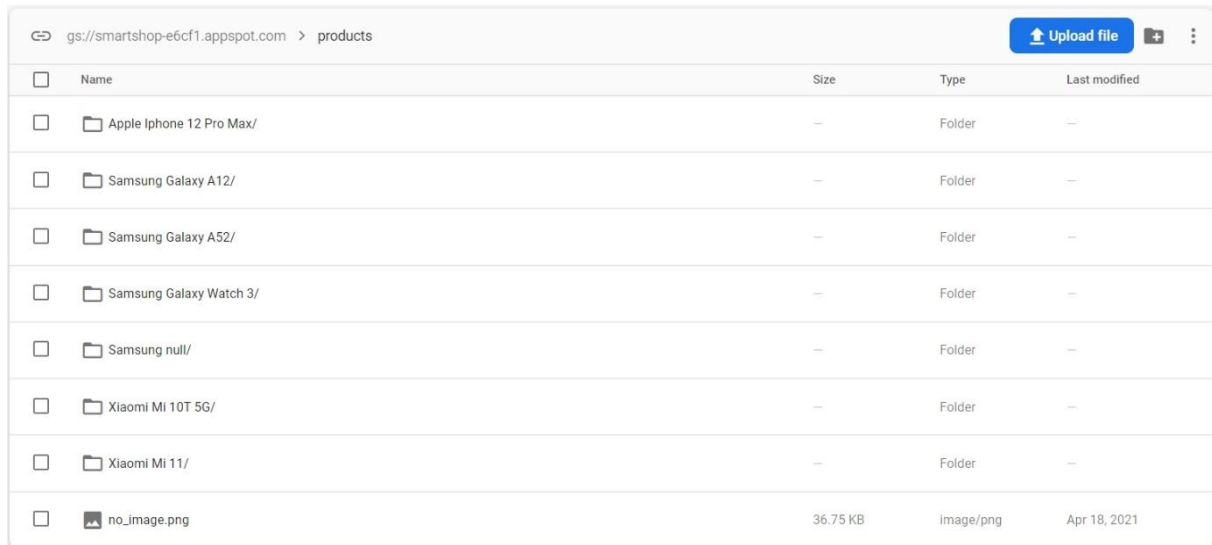
2.2.1. Cloud Storage

„Cloud Storage za Firebase namijenjen je programerima aplikacija koji trebaju pohranjivati i posluživati sadržaj koji stvaraju korisnici, poput fotografija ili videozapisa.“ (Storage, 2021.)

Cloud Storage je jednostavna i isplativa usluga za pohranu datoteka. U slučaju „SmartShop“ web aplikacije koristili smo Cloud Storage za pohranu fotografija koje se koristi na početnoj stranici u karuselu te pohranu fotografija svih proizvoda (Slika 7.).



Slika 7. Cloud Storage "SmartShop" web aplikacije



Slika 8. Organizacija spremanja fotografija proizvoda

Cloud Storage „SmartShop“ aplikacije je organiziran u dvije mape. U mapi *carousel* nalaze se fotografije karusela s početne stranice aplikacije. U mapi *products* nalaze se dodatne mape gdje svaka mapa predstavlja jedan proizvod (Slika 8.), a u kojima se na kraju i nalaze slike proizvoda. Ovdje se nalazi samo jedna fotografija *no_image.png* koja će biti prikazana u aplikaciji samo u slučaju da administrator nije dodao slike proizvoda prilikom dodavanja novog proizvoda.

2.3. Baza podataka

„Baza podataka je organizirana zbirka strukturiranih informacija ili podataka, obično pohranjenih elektroničkim putem u računalnom sustavu.“ (Oracle, "Database", 2021.)

Web aplikacije koriste baze podataka za spremanje informacija koje su relevantne za funkcioniranje određene web aplikacije. U primjeru „SmartShop“ web aplikacije, pošto je to e-commerce aplikacija, imamo potrebu spremati podatke o proizvodima, korisnicima, obavljenim narudžbama, listi želja korisnika i slično.

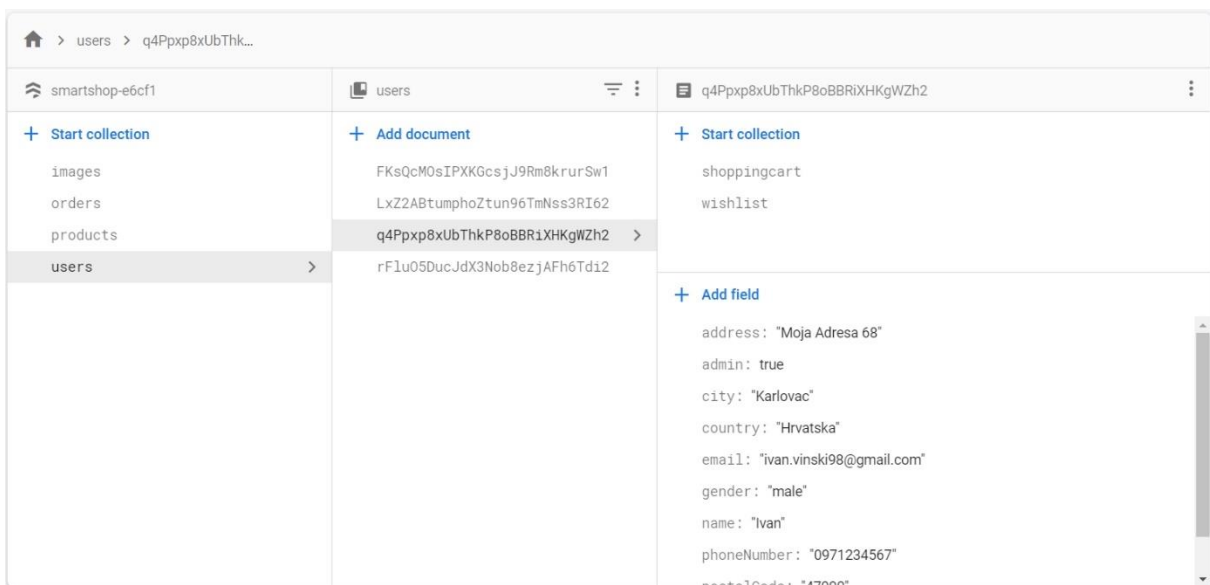
Postoje dvije vrste baze podataka, a to su relacijske (SQL) i nerelacijske (NoSQL) baze podataka. Glavna razlika između ove dvije vrste je ta da relacijske baze podataka pohranjuju podatke u tablice, dok nerelacijske baze pohranjuju podatke u obliku ključ/vrijednost, najčešće

u dokumentima ili na neki drugi način bez korištenja tablica. U izradi „SmartShop“ web aplikacije koristila se Firebase-ova Cloud Firestore NoSQL baza podataka.

2.3.1. Cloud Firestore

„Cloud Firestore je fleksibilna, skalabilna NoSQL baza podataka za razvoj mobilnih, web i server aplikacija.“ (Firestore, 2021.)

Podaci u ovoj bazi podataka se spremaju u dokumente koji sadrže polja u koja se dodaju vrijednosti. Ovi dokumenti se nalaze u kolekcijama, koje služe kao kontejneri za dokumente pomoću kojih se organiziraju podaci te vrše upiti nad njima. Dokumenti podržavaju razne tipove podataka, od onih jednostavnijih kao što su znakovni nizovi, brojevi, datum i vrijeme, reference i drugi, te kompliciranijih polja ili ugniježdeni objekti koji se nazivaju mape. Dokumenti također mogu sadržavati i podkolekcije i to je jedan od načina na koji se može organizirati baza podataka.



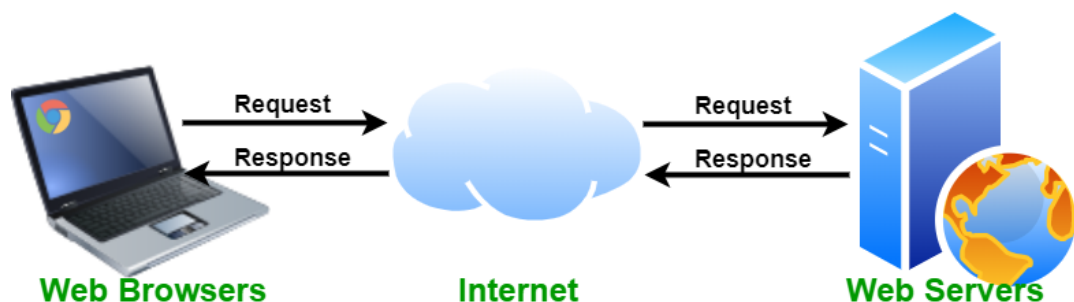
Slika 9. Baza podataka "SmartShop" web aplikacije

Baza podataka „SmartShop“ web aplikacije se sastoji od četiri kolekcije, a to su redom *images*, *orders*, *products* i *users* (prikazano na Slici 9.). U kolekciji *images* pohranjeni su podaci o slikama koje se koriste na početnoj stranici aplikacije. U kolekciji *orders* pohranjeni su podaci o obavljenim narudžbama kupaca gdje svaki dokument predstavlja jednu narudžbu. Dokumenti sadrže osnovne podatke o narudžbi, ali i podkolekciju *items* u kojoj svaki dokument predstavlja proizvod koji je naručen od strane kupca. Zatim imamo kolekciju *products* u kojoj se nalaze dokumenti koji predstavljaju proizvode. I na kraju imamo kolekciju *users* gdje su pohranjeni podaci o registriranim kupcima te administratorima web aplikacije. Svaki kupac, odnosno dokument može sadržavati, ali i ne mora, podkolekcije *shoppingcart* i *wishlist*. Ove podkolekcije će biti prikazane samo ako je kupac dodao barem jedan proizvod u košaricu (*shoppingcart*) ili listu želja (*wishlist*).

2.4. Web server

„Server (poslužitelj) je vrsta računala ili uređaja na mreži koji upravlja mrežnim resursima.“ (Beal, 2021.)

„Web serveri su računala koja isporučuju (ili opslužuju) web stranice. Svaki web server ima IP adresu i eventualno naziv domene. Postoje mnoge softverske aplikacije za web poslužitelje, uključujući softver javne domene i komercijalne pakete.“ (Beal, 2021.)



DG

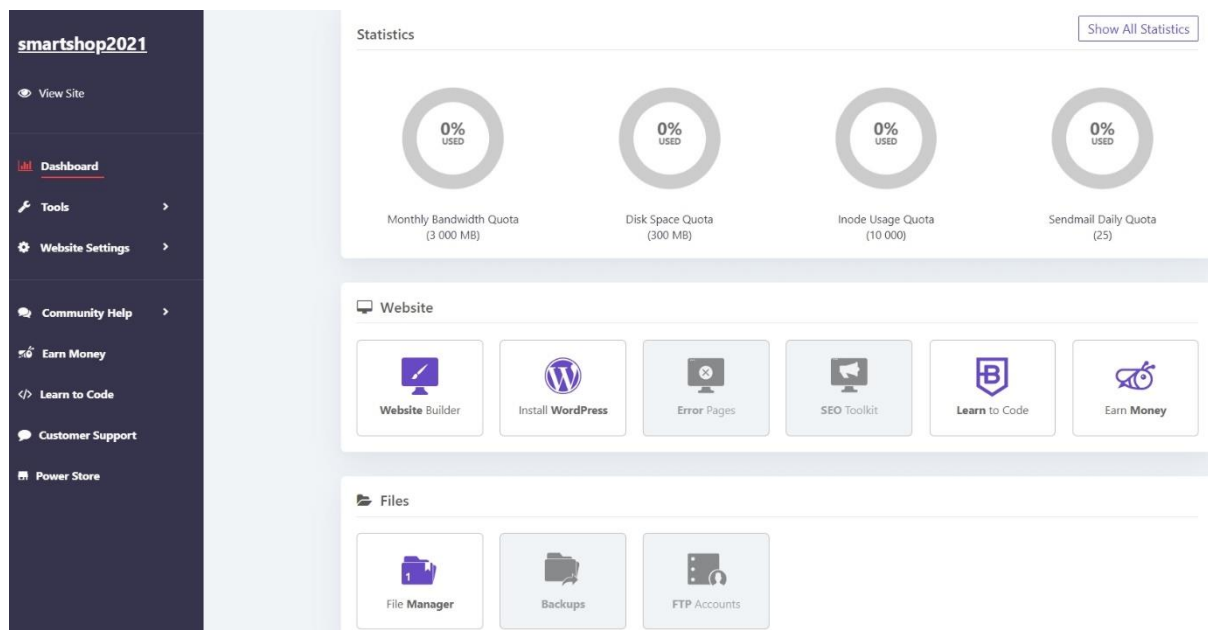
Slika 10. Primjer Web servera - Autor: [vitteshagupta](#) – License: [CC BY-SA 2.0](#)

Slika 10. nam prikazuje kako funkcionira razmjena podataka između klijentskog računala i web servera. Klijent preko web preglednika šalje zahtjev (engl. request) putem interneta prema web serveru, koji zatim nakon obrade zaprimljenih podataka šalje odgovor (engl. response) nazad prema klijentu.

Da bi se web aplikaciji moglo pristupiti putem weba potrebno je programski kod aplikacije objaviti na nekom serveru. Kako bi to bilo moguće potreban nam je web hosting servis. Postoji mnogo različitih pružatelja usluga web hostinga, od onih besplatnih, preko jeftinih pa do onih skupih. Zavisno o cijeni koju plaćamo za usluge web hostinga dobivamo različite mogućnosti. U skupljim hosting paketima dobivamo veće brzine, više memorije za pohranu, bolju zaštitu podataka na serveru itd. Za potrebe ovog završnog rada dovoljna nam je usluga besplatnog web hostinga. „SmartShop“ web aplikacija koristi usluge besplatnog web hosting servisa pod nazivom „000webhost“.

2.4.1. 000webhost

„000webhost“ pruža usluge web hostinga. Nude se četiri različita paketa za odabir, od kojih je prvi paket potpuno besplatan. Besplatni paket uključuje jednu web stranicu, 300 MB diskovnog prostora, ograničenu propusnost (Internet prometa) do 3 GB te besplatnu Cpanel upravljačku ploču hosting računa. Na Slici 11. prikazana je početna stranica Cpanel-a 000webhost-a. „SmartShop“ web aplikacija nalazi se na sljedećem linku: <https://smartshop2021.000webhostapp.com/#/>.



Slika 11. Cpanel 000webhost-a

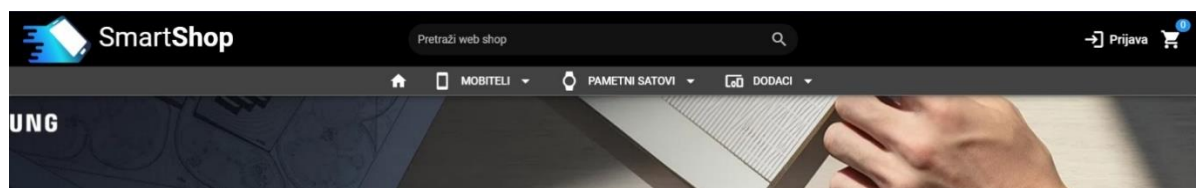
3. Opis elemenata i funkcionalnosti web aplikacije

3.1. Main layout

Glavni raspored web aplikacije sastoji se od zaglavlja (engl. header), spremnika stranica (engl. page container) i podnožja (engl. footer). U sljedećim potpoglavljima bit će opisani svaki od prethodno navedenih elemenata.

3.1.1. Header

Na *Slici 12.* prikazano je zaglavlje aplikacije. Zaglavlje se sastoji od alatne trake (engl. toolbar) i glavne navigacije u obliku horizontalnog izbornika.



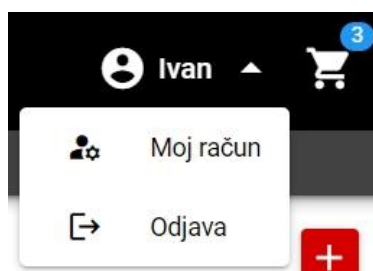
Slika 12. Header "SmartShop" aplikacije

U lijevom uglu alatne trake nalazi se logo i naziv web trgovine. Bez obzira na kojoj stranici se nalazili, klikom na logo ili naziv vraćamo se na početnu stranicu. U sredini se nalazi pretraživač pomoću kojeg možemo pretražiti proizvode na web trgovini prema nazivu proizvoda. Klikom na polje pretraživanja, ono poprimi bijelu boju, a ikona povećala se promjeni u ikonu križića što možemo vidjeti na *Slici 13.* Klikom na ikonu križića sav tekst koji je unesen u polje unosa pretraživača se briše. Nakon unosa pojma kojeg želimo pretražiti, pritiskom tipke „ENTER“ na tipkovnici sustav nas prebacuje na novu stranicu s listom pronađenih proizvoda.



Slika 13. Polje za unos pretraživanja

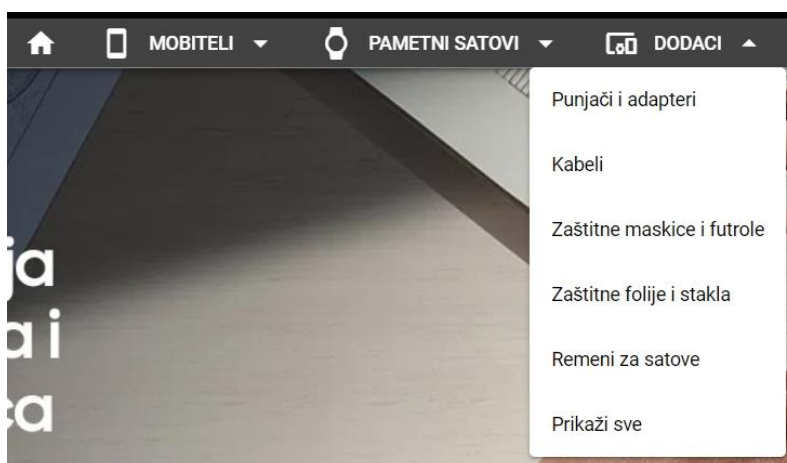
U desnom kutu alatne trake se nalaze dva gumba. Gumb „Prijava“ nas vodi na stranicu za prijavu. Nakon što se korisnik prijavi u sustav (ili registrira ukoliko je novi korisnik), gumb „Prijava“ postaje padajući gumb (engl. dropdown button), a tekst gumba se promijeni u ime korisnika kao što možemo vidjeti na Slici 14. U izborniku padajućeg gumba možemo birati između dvije opcije. Prva opcija „Moj račun“ vodi nas na podatke o trenutno prijavljenom korisniku, a druga opcija „Odjava“ odjavljuje trenutno prijavljenog korisnika te nas sustav vodi na stranicu za prijavu.



Slika 14. Padajući gumb korisnika

Zadnji gumb u alatnoj traci je gumb košarice. Ovaj gumb je predstavljen kao ikona kolica za kupovinu, što je vrlo intuitivno te nije potreban nikakav tekst. Korisnik pritiskom na taj gumb dolazi do stranice košarice, na kojoj se na kraju i obavlja narudžba. Gumb košarice također u gornjem desnom kutu sadrži i značku. Značka nam govori koliko se proizvoda trenutno nalazi unutar naše košarice. Ukoliko u košarici nemamo niti jedan proizvod biti će prikazan broj '0' što možemo vidjeti na primjeru Slike 12., dok nam Slika 14. prikazuje da se u košarici trenutno nalaze '3' proizvoda.

Ispod alatne trake nalazi se glavna navigacija koja je izvedena u obliku horizontalnog izbornika. Navigacija je centrirana tako se uvijek nalazi u sredini ekrana neovisno o širini ekrana. Prvi gumb u navigaciji je ikona kućice pomoću koje se vraćamo na početnu stranicu. Zatim slijede tri padajuća izbornika kod kojih možemo izabrati jednu od ponuđenih opcija ili odabrati opciju „Prikaži sve“ koja se nalazi na dnu liste (Slika 15.).



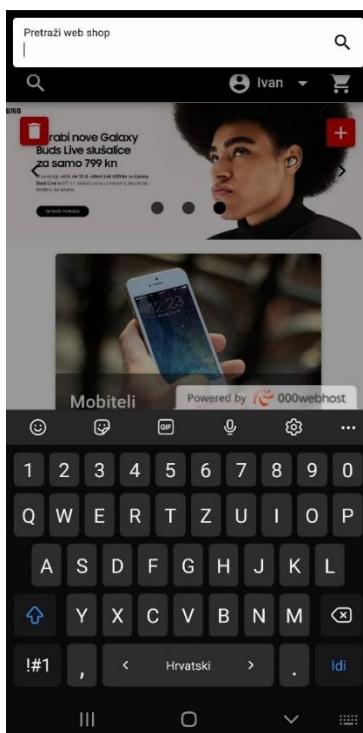
Slika 15. Horizontalna navigacija sa padajućim izbornicima

Zaglavlje na mobilnim uređajima je ponešto drugačije od zaglavlja na desktop računalima, no ima istu funkcionalnost. Došlo je do promjene rasporeda što se može vidjeti na *Slici 16*.

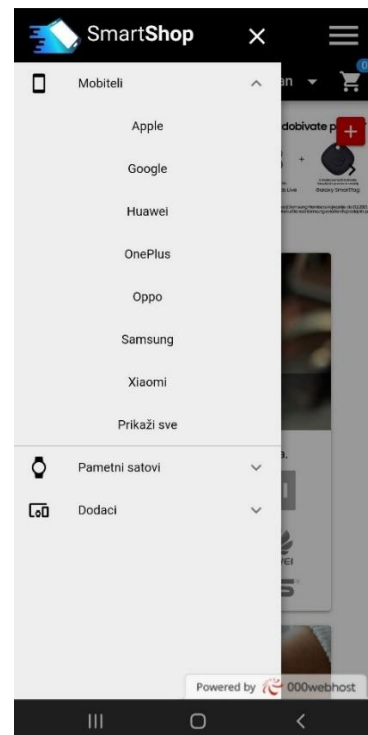


Slika 16. Header na mobilnim uređajima

Pretraživač se također promijenio. Više nije prikazano cijelo polje za pretraživanje već samo gumb sa ikonom povećala. Pritiskom na gumb, s vrha zaslona spušta nam se dijalog s poljem za unos pretraživanja, koje nam odmah bude u fokusu, odnosno ima uključenu opciju „autofocus“. To znači da nam se nakon pritiska na gumb pretraživača, prikazuje i tipkovnica od mobitela te smo odmah spremni za unos teksta koji želimo pretraživati (*Slika 17*).



Slika 17. Polje za pretraživanje na mobilnim uređajima



Slika 18. Mobilna navigacija

Još jedan element se promijenio u odnosu na desktop prikaz, a to je navigacija. Horizontalna navigacija je u potpunosti nestala, a pojavio se „hamburger“ meni. Ova navigacija je dobila takvo ime zbog toga što je ikona za otvaranje ove navigacije slična hamburgeru. Pritiskom na gumb „hamburgera“, s lijeve strane nam izlazi vertikalna navigacija s tri padajuća izbornika (*Slika 18*). Na vrhu se nalazi logo i naziv te gumb za zatvaranje navigacije. Ova navigacija se također može prikazati i sakriti povlačenjem (engl. swipe) udesno i ulijevo.

3.1.2. Page container

Page container je središnji dio aplikacije koji se smjestio između zaglavlja i podnožja. U njemu nam se prikazuju različite stranice. Dok su zaglavlje i podnožje kroz cijelu aplikaciju isti, ovo je dio koji se mijenja kretanjem kroz navigaciju web aplikacije.

3.1.3. Footer

Podnožje je podijeljeno na tri sekcije (prikazano na *Slici 19.*). U lijevoj sekciji se nalazi sekundarna navigacija u obliku vertikalnog izbornika. Imamo tri opcije, a to su: *O nama*, *Dostava* i *Plaćanje*. Klikom na neku od tih opcija, sustav nas prebacuje na njihov sadržaj. U sredini se nalaze logo i naziv web aplikacije, te klikom na njih sustav nas vraća na početnu stranicu. Na kraju, u desnoj sekciji imamo tri gumba s ikonama Facebook-a, Instagram-a i YouTube-a koje nas vode na navedene društvene mreže. U slučaju mobilnog prikaza, srednja sekcija je sakrivena.



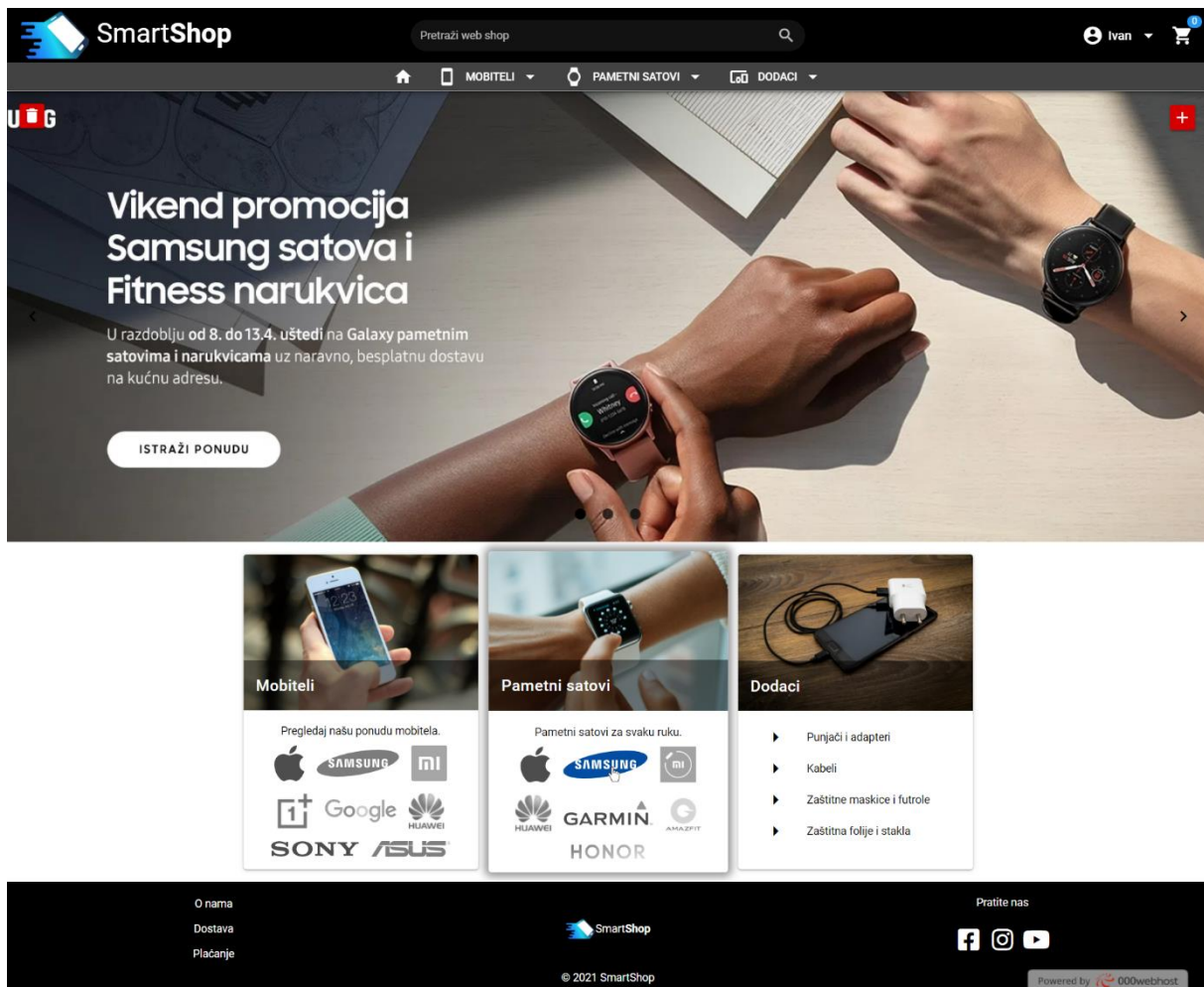
Slika 19. Footer "SmartShop" web aplikacije

3.2. Opis stranica

3.2.1. Početna stranica

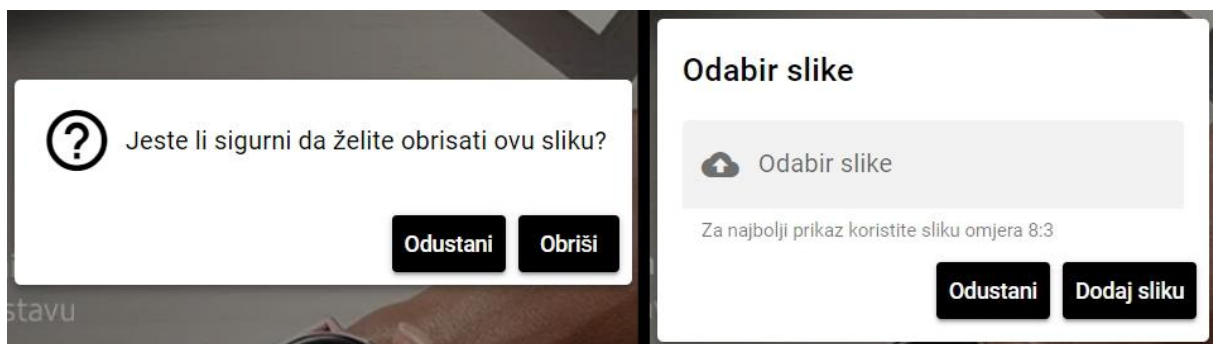
Početna stranica se sastoji od dva dijela što možemo vidjeti na *Slici 20.* Prvi dio je karusel sa fotografijama, odnosno dijaprojekcija (engl. slideshow) fotografija. To je prva stvar koju korisnik primijeti dolaskom na početnu stranicu web aplikacije. One nam prikazuju trenutne akcije, promocije ili općenito bilo kakve obavijesti koje administrator web aplikacije želi prikazati. Svakih 5 sekundi fotografije se automatski izmjenjuju (autoplay) osim u slučaju kada se pokazivač miša nalazi iznad karusela. Tada se autoplay pauzira sve dok se pokazivač ne pomakne na neko drugo područje web stranice. Korisnik se također može ručno kretati kroz fotografije pritiskom na gumb strelica koje se nalaze na lijevoj i desnoj strani, pritiskom na kružice koji se nalaze na dnu karusela ili povlačenjem („svajpanjem“) ulijevo i udesno.

U drugom dijelu koji se nalazi ispod karusela imamo tri kartice, od kojih svaka predstavlja jednu kategoriju proizvoda. Svaka kartica se sastoji od dvije sekcije, gdje se u prvoj nalazi fotografija i tekst, a u drugoj dodatne opcije. Kartice imaju hover efekt takav da poprime malo uvećanje (zoom) i sjenu oko kartice, što se može vidjeti na srednjoj kartici „*Pametni satovi*“. Klikom na područje fotografije sustav nas prebacuje na prikaz svih proizvoda iz te kategorije, a klikom na logo proizvođača (ili u kategoriji „*Dodaci*“ na neku od potkategorija) biti će prikazani proizvodi iz samo te potkategorije. Također možemo primijetiti da je svaki logo u sivim tonovima, a poprime boju samo kada se pokazivačem nalazimo iznad loga što možemo vidjeti u slučaju loga *Samsunga* u srednjoj kartici.

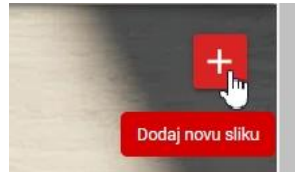


Slika 20. Početna stranica

Ukoliko je prijavljeni korisnik ujedno i administrator web aplikacije on ima dvije dodatne mogućnosti, a to su brisanje postojećih ili dodavanje novih slika u karusel. Ukoliko korisnik nije administrator ovi gumbi mu neće biti prikazani. Gumb za brisanje nalazi se u gornjem lijevom kutu s ikonom koša za smeće, dok se gumb za dodavanje nove slike nalazi u gornjem desnom kutu s ikonom plusa. Oba gumba imaju hover efekt opis alata (engl. tooltip) što možemo vidjeti na *Slici 22. Slika 21.* nam prikazuje dijaloške okvire za brisanje trenutne ili dodavanje nove slike.



Slika 21. Dijalozi za brisanje trenutne i dodavanje nove slike



Slika 22. Tooltip

Prikaz na mobilnim uređajima je neznatno drugačiji. Karusel fotografija je responzivan stoga se on samo prilagodio veličini ekrana dok su se kartice poredale jedna ispod druge.

3.2.2. Prijava/Registracija

Stranice *Prijave* i *Registracije* su dosta slične stoga su opisane u istom poglavlju. Na *Slici 23.* prikazana je forma za prijavu, dok je na *Slici 24.* prikazana forma za registraciju.

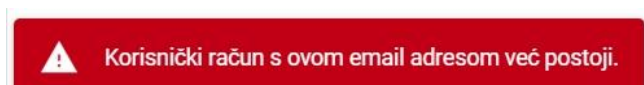
Slika 23. Forma za prijavu

Slika 24. Forma za registraciju

Za prijavu je potrebno upisati email adresu i lozinku te pritiskom na gumb *Prijava* korisnik se prijavljuje u sustav. Ukoliko su upisani podaci netočni (nepostojeća email adresa, netočna lozinka...) sustav nas pomoću notifikacije u donjem dijelu ekrana o tome upozorava (*Slika 25.*). Prilikom registracije potrebno je unijeti ime, prezime, email adresu i lozinku te pritiskom na gumb *Registriraj se* korisnik se registrira u sustav. Ukoliko se prilikom registracije unese email adresa koja već postoji u sustavu, sustav će nas o tome upozoriti pomoću notifikacije (*Slika 26.*).



Slika 25. Notifikacija o greški prilikom prijave



Slika 26. Notifikacija o greški prilikom registracije

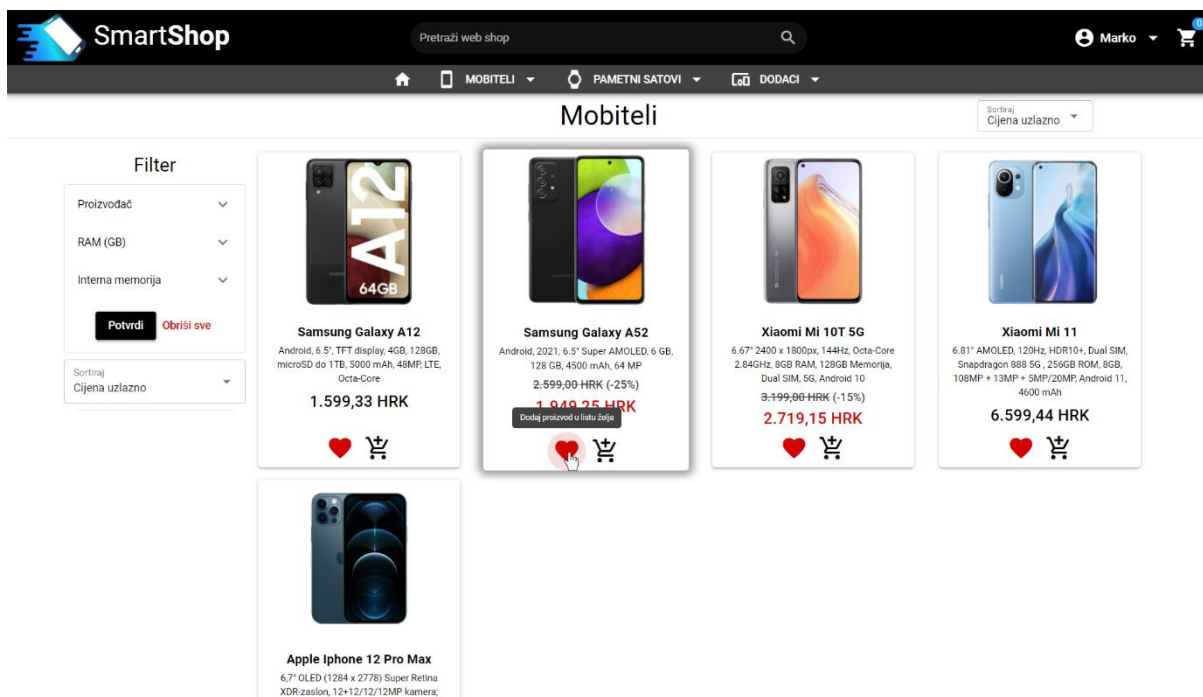
3.2.3. Prikaz proizvoda

Stranice za prikaz proizvoda su uz stranicu *Košarice* jedne od kompliciranijih stranica. Postoje četiri stranice koje nam služe za prikaz proizvoda, a to su: *Mobiteli* (engl. *Phones*), *Pametni satovi* (engl. *Smart watches*), *Dodaci* (engl. *Accessories*) i *Rezultati pretrage* (engl. *Search results*). Sve četiri stranice su vrlo slične uz samo neke minimalne razlike. Iz tog razloga biti će opisana samo stranica *Mobiteli* te će biti navedene razlike između ostalih stranica. Na *Slici 27.* prikazana je stranica *Mobiteli*.

Na vrhu se nalazi naslov stranice koji je centriran. Desno od naslova nalazi se element za sortiranje proizvoda koji je izveden kao padajući izbornik. Proizvodi se mogu sortirati prema cijeni (silazno ili uzlazno), abecedno ili prema najvećem popustu (gdje se proizvodi s najvećim popustom prikazuju prvi).

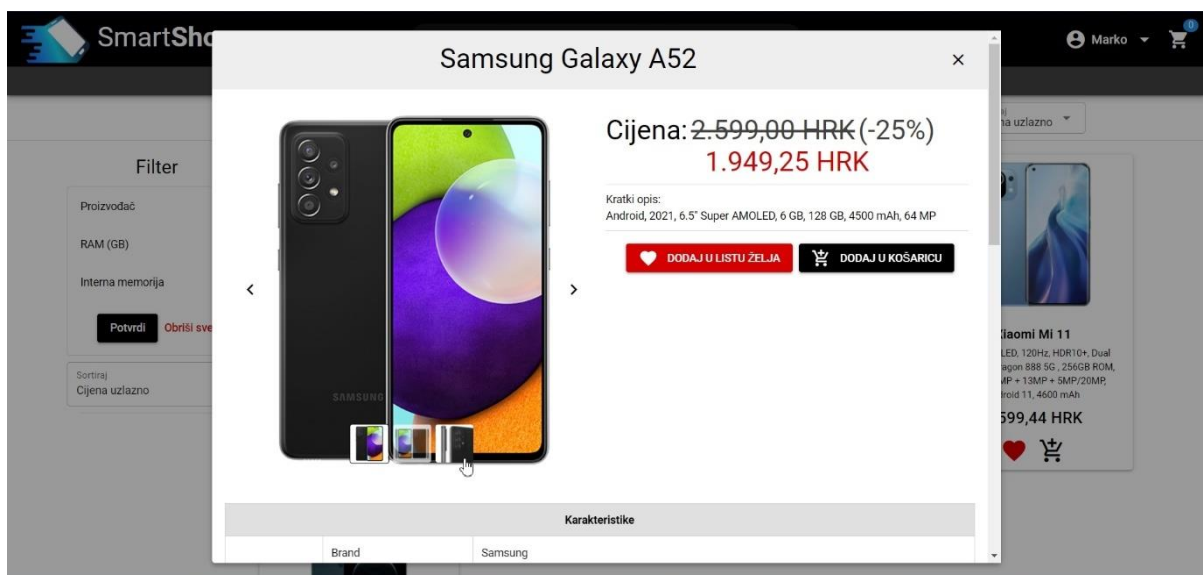
Ispod naslova nalaze se dvije sekcije. U lijevoj, užoj sekciji nalazi se filter proizvoda, a ispod njega još jedan izbornik za sortiranje koji je izveden isto kao i prethodni. Filter proizvoda je izveden kao lista sa proširivim stavkama. Klikom na neku od stavki u listi, ona se proširuje te korisnik ima mogućnost odabrati opcije po kojima želi filtrirati proizvode. Nakon odabira opcija, klikom na gumb „*Potvrdi*“ lista proizvoda će biti filtrirana. Klikom na gumb „*Obriši sve*“, sve opcije filtriranja biti će uklonjene. Sortiranje radi isto kao i sortiranje na vrhu stranice stoga ga nije potrebno posebno opisivati.

Desno od filtera nalazi se lista proizvoda izvedena u obliku kartica. Kartice su podijeljene na tri sekcije. U gornjem dijelu kartice se nalazi slika proizvoda. U središnjem dijelu kartice imamo naziv proizvoda koji je podebljan. Ispod naziva se nalazi kratki opis proizvoda. Zatim slijedi cijena proizvoda koja je također podebljana. Ukoliko je proizvod na popustu, stara cijena je precrtana i pored nje u zagradi je napisan iznos popusta u postotku. Cijena s uračunatim popustom se nalazi ispod stare cijene te je istaknuta crvenom bojom. Na dnu kartice se nalaze dva gumb. Gumb s ikonom srca služi za dodavanje proizvoda u listu želja, dok gumb s ikonom košarice služi za dodavanje proizvoda u košaricu. Kako bi korisnici znali točno značenje ikona, dodan je tooltip s opisom značenja gumba koji se pojavljuje kada se pokazivačem miša prijeđe preko gumba. Svaka kartica ima hover efekt uvećanja i dodavanja sjene oko kartice. Ovisno o širini ekrana s kojeg korisnik posjećuje web aplikaciju prikazivat će se četiri, tri ili dva proizvoda u istom retku. Na ekranima koji su manji od 1023 piksela, filter će se prikazati na vrhu, a ispod njega lista proizvoda.



Slika 27. Stranica "Mobiteli"

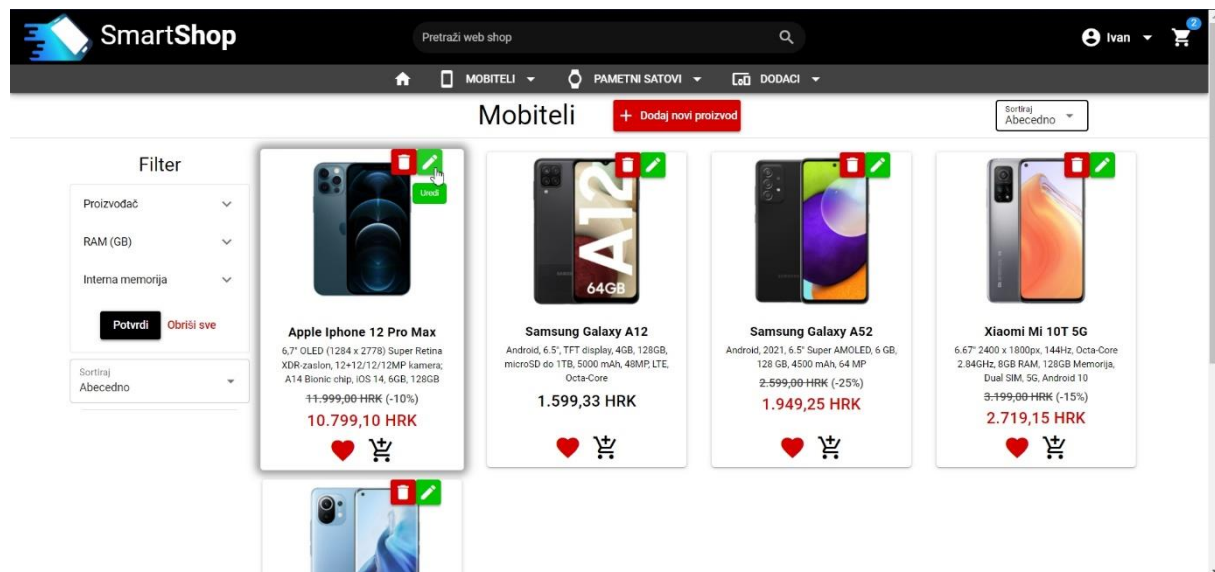
Pritiskom na bilo koji dio kartice osim na gumbе za dodavanje proizvoda u listu želja i košaricu otvara se dijaloški okvir s detaljnijim informacijama o proizvodu (Slika 28.).



Slika 28. Dijaloški okvir za prikaz detaljnih informacija o proizvodu

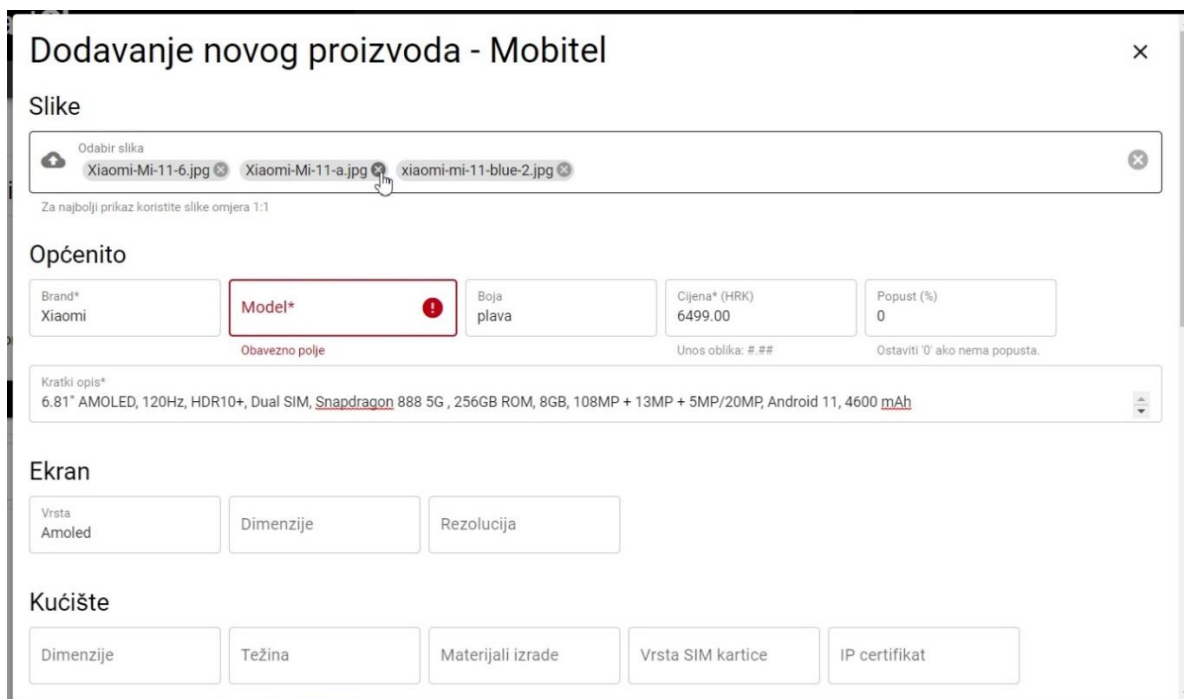
Dijaloški okvir s detaljnim informacijama o proizvodu sastoji se od tri sekcije. U prvoj sekciji koja je smještena na vrhu nalazi se naziv proizvoda i gumb s ikonom križića koji nam služi za zatvaranje dijaloškog okvira. Dijaloški okvir također je moguće zatvoriti pritiskom na bilo koje mjesto izvan okvira. U drugoj sekciji s lijeve strane se nalaze slike proizvoda koje je moguće izmjenjivati pritiskom na strelice ili male sličice (engl. thumbnails). Desno od slike proizvoda nalaze se cijena proizvoda, kratki opis proizvoda i gumbi za dodavanje proizvoda u listu želja ili košaricu. U zadnjoj sekciji na dnu se nalazi tablica sa svim karakteristikama proizvoda.

Kod prikaza na mobilnim uređajima samo će se srednja sekcija promijeniti, odnosno doći će do promjene rasporeda. Slika će biti na vrhu, a cijena, kratki opis i gumbi ispod nje.



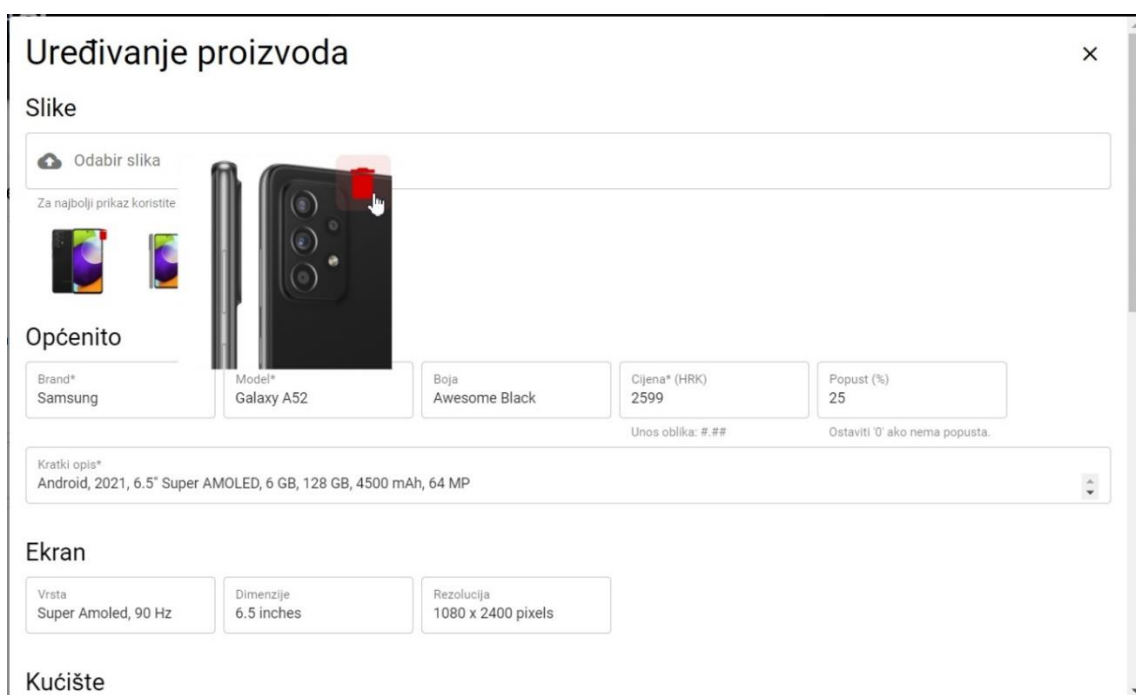
Slika 29. Stranica "Mobiteli" ukoliko je prijavljeni korisnik administrator

Ukoliko je prijavljeni korisnik ujedno i administrator, on će imati par dodatnih mogućnosti što možemo vidjeti na Slici 29. Administratoru će na vrhu pored naslova biti prikazan gumb *Dodaj novi proizvod*. Pomoću ovog gumba dodajemo novi proizvod u kategoriju *Mobiteli*. Ako bi se nalazili na stanici *Pametni satovi* ili *Dodaci*, onda bi se proizvod dodavao u te kategorije. Klikom na gumb otvara se dijaloški okvir za dodavanje novog proizvoda (Slika 30.). Dijaloški okvir sastoji se od raznih inputa, odnosno polja za unos. Prvi input služi za dodavanje slika proizvoda. Moguće je u jednom unosu dodati više slika. Ukoliko smo odabrali krivu sliku, pomoću križića koji se nalazi pored naziva slike moguće je ukloniti samo tu sliku. Ako iz nekog razloga želimo ukloniti sve slike, to možemo učiniti pomoću križića koji se nalazi skroz desno na kraju inputa. Ostala polja služe za unos ostalih podataka i karakteristika proizvoda od kojih su neki obavezni (označeni sa zvjezdicom). Ukoliko obavezna polja nisu ispunjena, nećemo moći spremiti novi proizvod. Nakon unosa podataka o proizvodu pomoću gumba *Spremi* koji se nalazi na dnu dijaloškog okvira dodajemo novi proizvod. Pored gumba *Spremi* nalazi se i gumb *Odustani* ako želimo odustati od dodavanja proizvoda.



Slika 30. Dijaloški okvir za dodavanje novog proizvoda

Kod svakog proizvoda u listi, možemo vidjeti dva gumba koja se nalaze u gornjem desnom kutu kartice. Crveni gumb služi za brisanje proizvoda. Klikom na njega otvara se dijaloški okvir za potvrdu brisanja odabranog proizvoda. Zeleni gumb služi za uređivanje proizvoda. Klikom na zeleni gumb otvara se isti dijaloški okvir koji služi za dodavanje proizvoda, samo su u ovom okviru polja za unos već ispunjena s podacima o proizvodu. Slika 31. nam prikazuje dijaloški okvir za uređivanje proizvoda. Možemo primijetiti da se ovdje ispod polja za unos slika, nalazi pregled već dodanih slika. Prelaskom pokazivačem miša preko slika, one se uvećavaju te imamo opciju za brisanje slike.

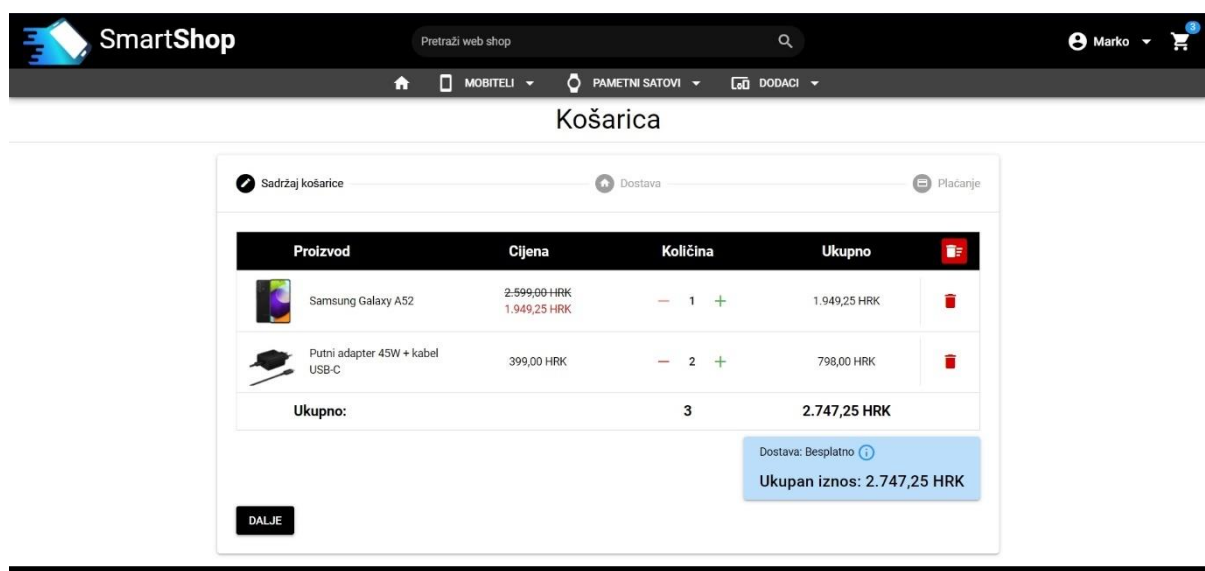


Slika 31. Dijaloški okvir za uređivanje proizvoda

3.2.4. Košarica

Na stranici *Košarica* (prikazano na *Slici 32.*) nalazi se okvir s nizom od tri koraka koje korisnik mora pratiti kako bi uspješno obavio narudžbu. Ova komponenta se naziva steper (engl. stepper).

U prvom koraku imamo sadržaj košarice. Ovdje nam je prikazana tablica s pet stupaca. Prvi stupac je *Proizvod* u kojem se nalazi slika i naziv proizvoda. Zatim imamo cijenu proizvoda. U trećem stupcu se nalazi količina proizvoda. Količinu možemo mijenjati s gumbovima minus (crveni) i plus (zeleni). U četvrtom stupcu je iskazana ukupna cijena proizvoda na način da je cijena proizvoda pomnožena s količinom. U zadnjem stupcu u zaglavlju se nalazi gumb za brisanje svih proizvoda iz košarice, a pomoću gumba koji se nalazi u retku s proizvodom brišemo samo taj proizvod i sve njegove količine. U zadnjem retku tablice je iskazana ukupna količina svih proizvoda i njihova cijena. Ispod tablice na desnoj strani se nalazi okvir s plavom pozadinom gdje je iskazana cijena dostave te ukupan iznos narudžbe kada se pribroji i dostava.



Slika 32. Stranica "Košarica"

U drugom koraku potrebno je unijeti podatke o adresi dostave (*Slika 33.*). Ukoliko su ti podaci već uneseni na stranici *Moj račun* oni će se automatski ispuniti, no moguće ih je i izmijeniti ukoliko želimo da se proizvodi dostave na drugu adresu. Ovdje su sva polja obavezna i ne možemo nastaviti dalje dok ne ispunimo sve podatke.

Slika 33. Drugi korak: Dostava

U zadnjem koraku biramo način plaćanja (Slika 35.). Imamo tri opcije, a to su: *Plaćanje pouzećem*, *Kartično plaćanje* i *Plaćanje putem virmana ili opće uplatnice*. Sve dok ne izaberemo neku od ponuđenih opcija gumb *Plati* je onemogućen (engl. disabled) i ne možemo obaviti kupovinu. Nakon što smo odabrali opciju i kliknuli gumb *Plati* otvara se dijaloški okvir u kojem još jednom moramo potvrditi plaćanje (Slika 34.). Nakon potvrde sustav nas prebacuje na pregled obavljenih narudžbi.

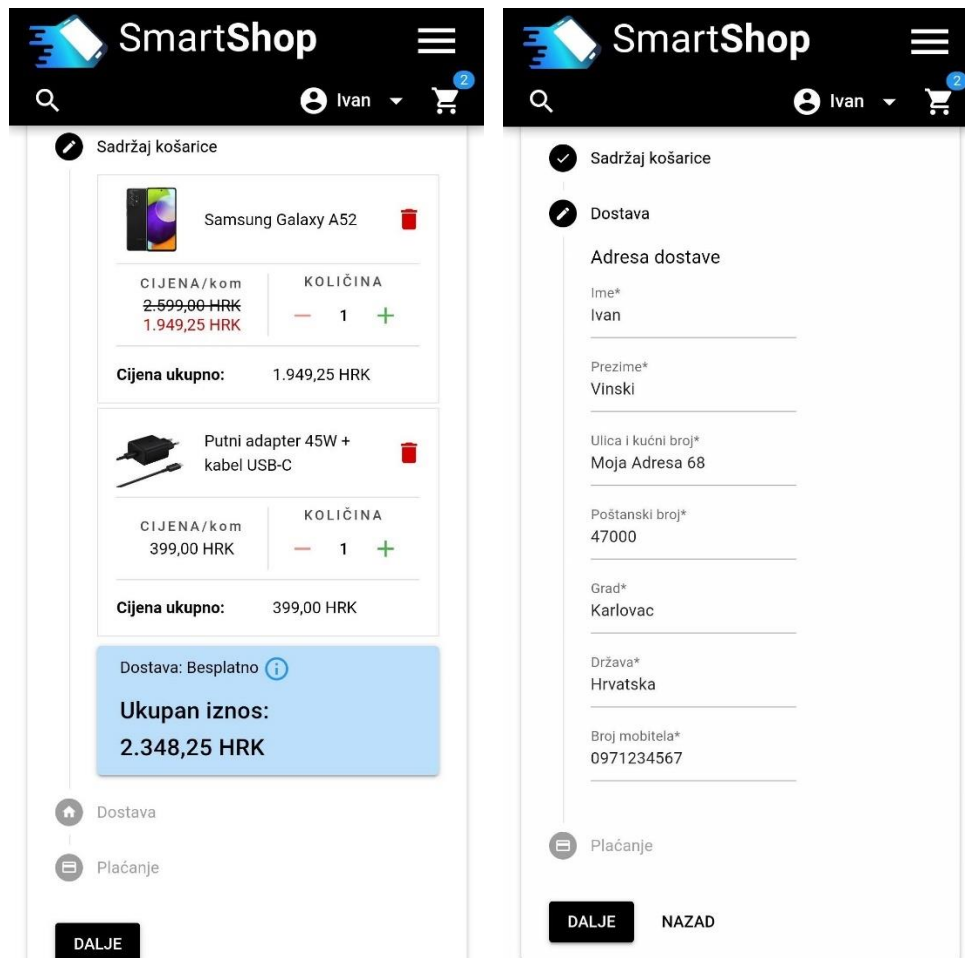


Slika 34. Potvrda plaćanja

Slika 35. Treći korak: Plaćanje

Kod mobilnog prikaza, horizontalni stepper postati će vertikalni stepper. U prvom koraku će se umjesto tablice s proizvodima prikazati kartice proizvoda, poredanih jedna ispod druge. Na ovim karticama prikazane su iste informacije kao i u tablici, samo na drugačiji način (Slika 36.).

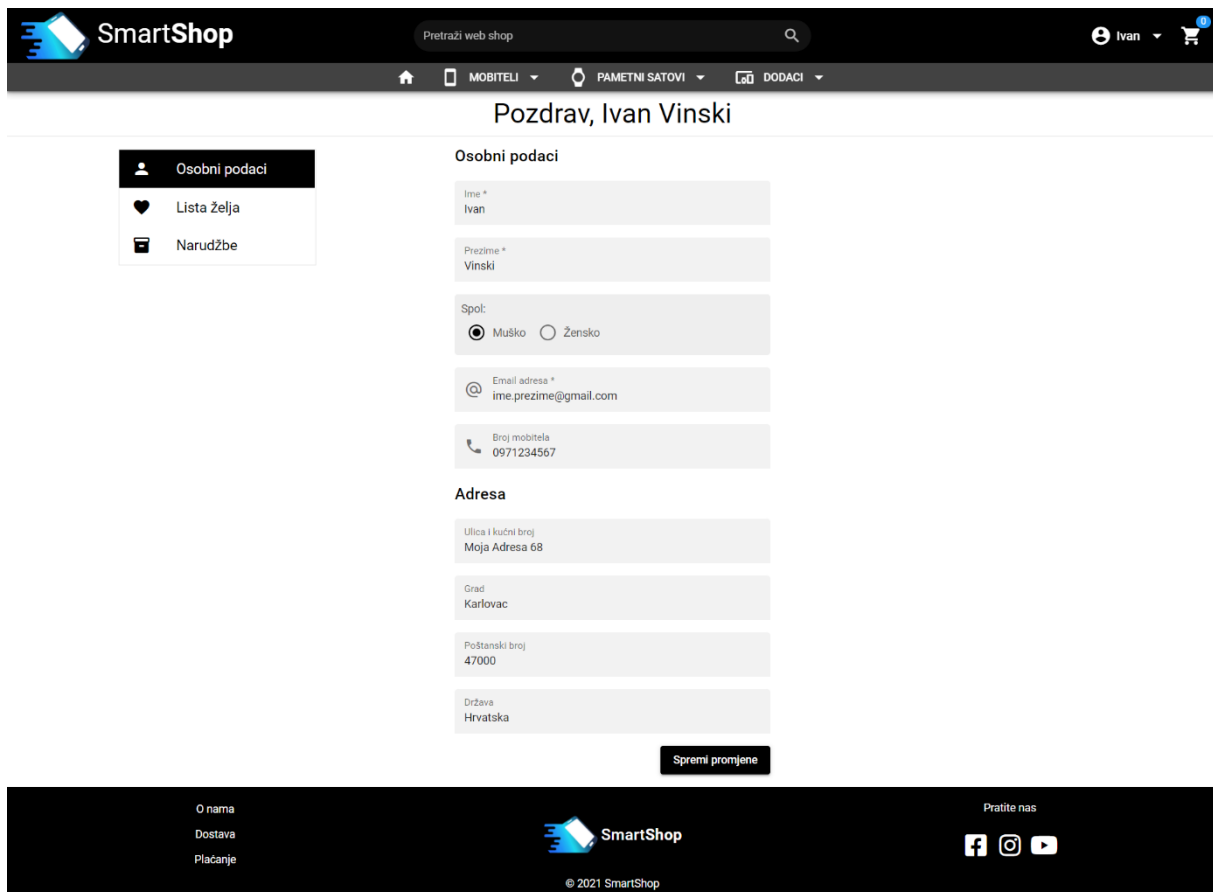
Drugi i treći koraci su isti kao i na desktop prikazu uz samo malu promjenu rasporeda elemenata.



Slika 36. Mobilni prikaz stranice "Košarica"

3.2.5. Moj račun

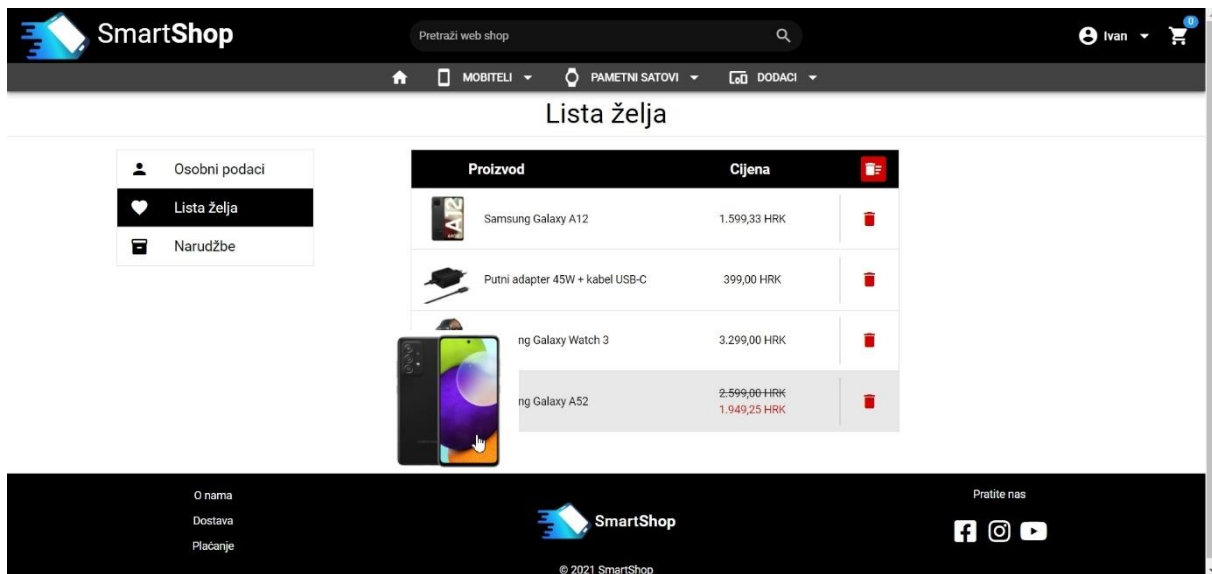
Na stranici *Moj račun* (prikazano na Slici 37.) imamo prikaz osnovnih podataka o korisniku. Ovoj stranici nije moguće pristupiti ukoliko korisnik nije prijavljen u sustav. Na samom vrhu nalazi se pozdravna poruka i ime prijavljenog korisnika. Ispod toga s lijeve strane nalazi se sekundarna navigacija izvedena u obliku vertikalnog izbornika. Trenutno aktivna opcija u navigaciji ima crnu pozadinu s bijelim tekstom, dok ostale opcije imaju bijelu pozadinu sa crnim tekstom. Desno od navigacije se nalaze osobni podaci korisnika i njegova adresa. Te podatke je moguće izmijeniti. Nakon što se unesu željene promjene potrebno je pritisnuti gumb *Spremi promjene* na dnu stranice kako bi se izmijenjeni podaci ažurirali u bazi podataka. Polja *Ime*, *Prezime* i *Email adresa* su označena zvjezdicom i to su obavezna polja koja ne smiju ostati prazna. Sustav neće dozvoliti spremanje podataka ako je neko od tih triju polja ostalo prazno. Jedina razlika između desktop i mobilnog prikaza je taj da navigacija dolazi na vrh stranice, dok se osobni podaci nalaze ispod nje.



Slika 37. Stranica "Moj račun"

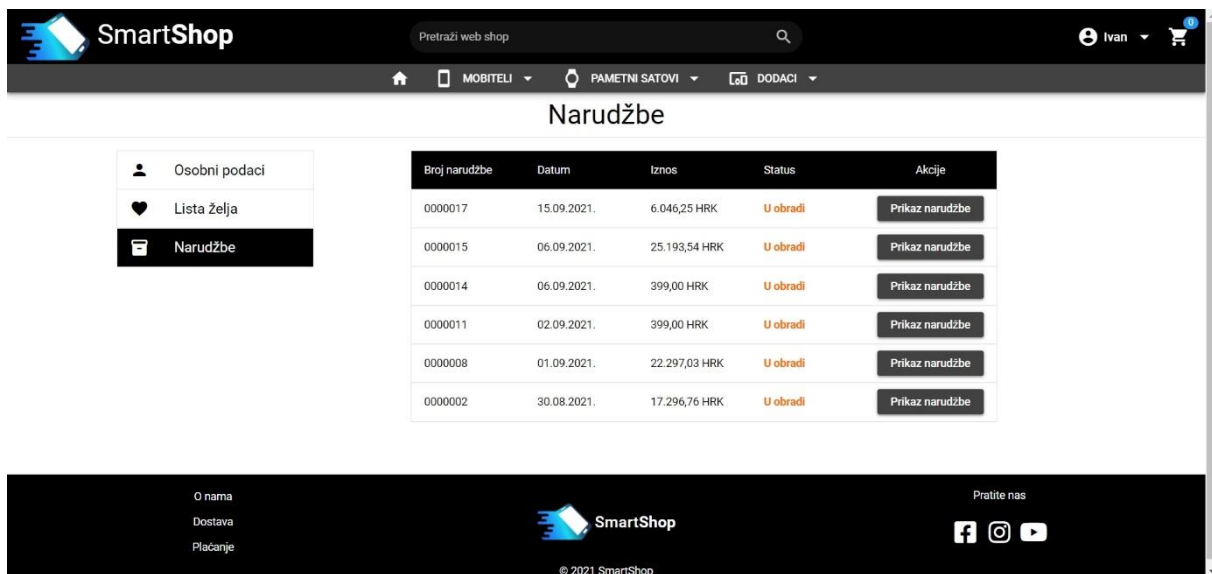
3.2.6. Lista želja

Na stranici *Lista želja* (prikazano na *Slici 38.*) nalazi se tablica, odnosno lista proizvoda koje je korisnik dodao u listu. Zaglavlje tablice se sastoji od tri sekcije: *Proizvod*, *Cijena* i gumb za brisanje svih proizvoda iz liste želja. Redak tablice sadrži sliku proizvoda, naziv proizvoda, cijenu te gumb za brisanje pojedinog proizvoda. Gumb je od ostatka retka odvojen vertikalnom crtom odnosno separatorom. Slika proizvoda se uveća ukoliko se pokazivač miša postavi iznad nje. Ukoliko je proizvod koji se nalazi u listi želja na popustu, stara cijena će biti precrtana, a nova cijena će biti istaknuta crvenom bojom. Mobilni prikaz liste želja isti je kao i desktop prikaz. Jedino dolazi do promjene rasporeda navigacije i tablice kao i kod stranice *Moj račun*.



Slika 38. Stranica "Lista želja"

3.2.7. Narudžbe



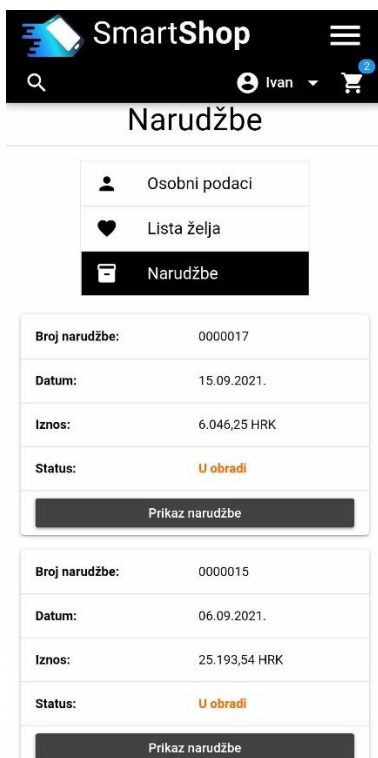
Slika 39. Stranica "Narudžbe"

Na stranici *Narudžbe* (prikazano na *Slici 39.*) nalazi se popis svih obavljenih narudžbi. Prvi stupac tablice se odnosi na broj narudžbe. Drugi stupac je datum kada se narudžba izvršila. Zatim imamo iznos cjelokupne narudžbe. Sljedeći stupac nam govori o statusu u kojem se trenutno narudžba nalazi, a to može biti: *U obradi*, *Poslano* ili *Dostavljeno*. I za kraj imamo stupac *Akcije* u kojem se nalazi gumb za prikaz detalja pojedine narudžbe. Pritiskom na gumb otvara se dijaloški okvir s detaljima narudžbe koji se može vidjeti na *Slici 40.* Na vrhu se nalazi naslov *Detalji narudžbe*. Ispod naslova su općenite informacije o narudžbi kao što su broj, datum i status narudžbe. Zatim slijedi tablica s podacima o svakom pojedinom artiklu narudžbe. Ispod tablice nalaze se dodatni podaci o narudžbi, a to su *Adresa dostave*, *Način plaćanja* i *Dostava*.

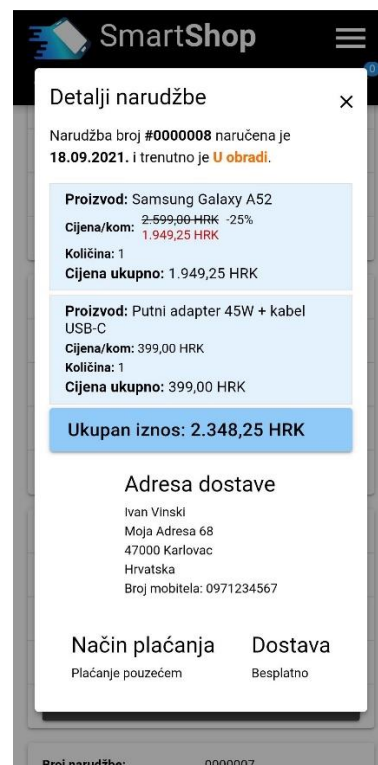


Slika 40. Detalji narudžbe

Mobilni prikaz stranice *Narudžbe* može se vidjeti na *Slici 41*. Umjesto u tablici, narudžbe su poredane jedna ispod druge u obliku kartice. Svaka kartica sadrži jednake podatke o narudžbi kao i kod desktop prikaza. U dijaloškom okviru detalja narudžbe na mobilnom prikazu jedino je popis artikala izveden na drugačiji način (*Slika 42.*). Artikli su poredani jedan ispod drugog u obliku kartice sa podacima o proizvodu.



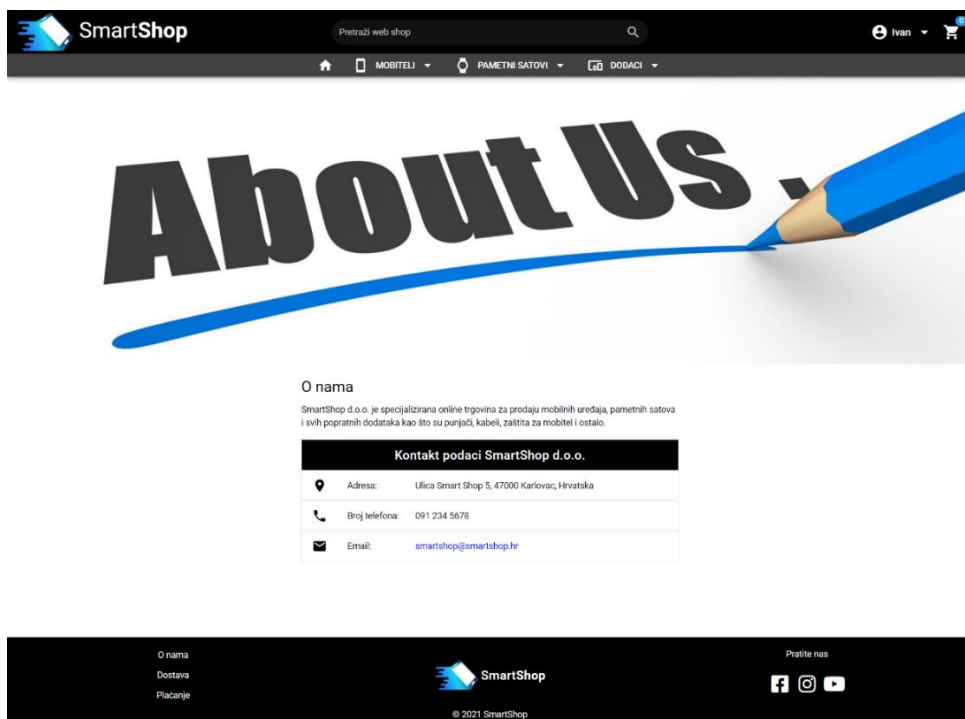
Slika 41. Mobilni prikaz stranice "Narudžbe"



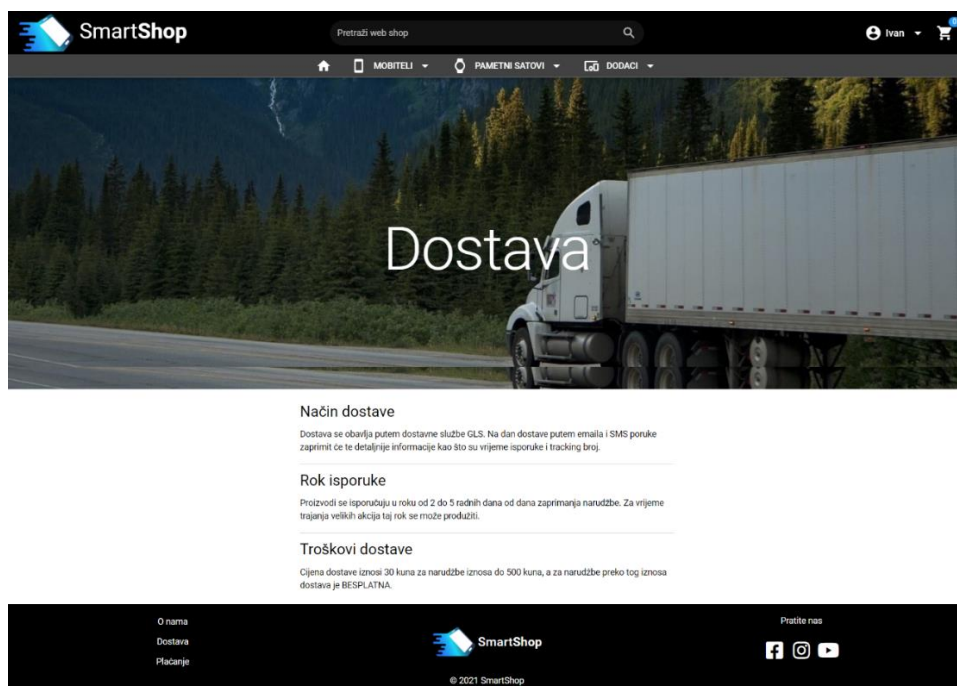
Slika 42. Mobilni prikaz Detalja narudžbe

3.2.8. O nama/Dostava/Plaćanje

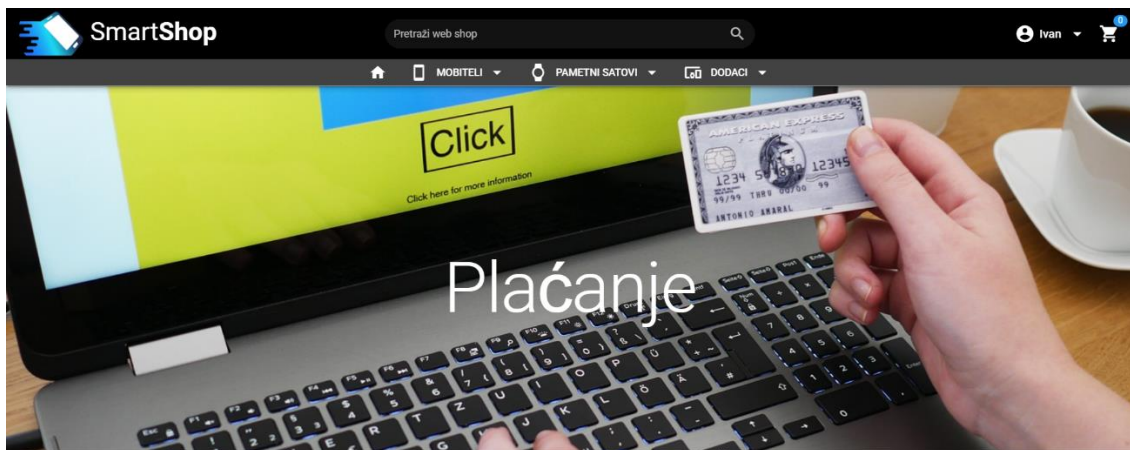
Stranice *O nama*, *Dostava* i *Plaćanje* (prikazane redom na *Slici 43.*, *44.* i *45.*) su tri vrlo slične stranice koje su ujedno i najjednostavnije stranice na cijeloj web aplikaciji. Stranice *Dostava* i *Plaćanje* ne sadrže nikakve interaktivne elemente, dok jedino stranica *O nama* ima samo jedan interaktivni element. Taj element je email adresa i klikom na nju otvara nam se zadana (engl. default) aplikacija za slanje email pošte. Na vrhu svake stranice nalazi se slika, a ispod nje neke informacije vezane uz web trgovinu.



Slika 43. Stranica "O nama"



Slika 44. Stranica "Dostava"



Plaćanje pouzecom

Plaćanje pouzecom omogućava Vam da iznos narudžbe platite prilikom dostave na adresu koju ste naveli kao adresu dostave. Uplatu možete izvršiti isključivo u gotovini, dostavljajući prilikom same isporuke. Ukoliko primatelj nije bio u mogućnosti primiti pošiljku na adresi dostave, dostavljač će ostaviti obavijest o prispjehu pošiljke.

Kartično plaćanje

Kupac naručenu robu plaća debitnim i kreditnim karticama služeći se T-com PayWay sustavom kartične internet naplate narudžbe. Plaćanje debitnim i kreditnim karticama omogućeno je za Visa, MasterCard, Maestro, Visa Premium i Diners kartice.



Plaćanje putem virmana ili opće uplatnice

Primatelj:
 SmartShop d.o.o.
 Ulica Smart Shop 5
 47000 Karlovac
 IBAN: HR0123456789876543210
 Poziv na broj: 02 - upisati broj narudžbe



Slika 45. Stranica "Plaćanje"

3.3. Programski kod

3.3.1. Povezivanje web aplikacije s firebase projektom

Web aplikaciju povezujemo s firebase projektom tako da u mapi *src/boot* stvorimo novu datoteku naziva *smartShopSetup.js* (Slika 47.). Da bi se ova datoteka uspješno izvela prilikom pokretanja aplikacije potrebno ju je uključiti unutar *quasar.conf.js* konfiguracijske datoteke na način da se doda u niz *boot* kako je prikazano na Slici 46.

Prve četiri linije koda služe za uvoz firebase sustava, odnosno sustava za autentifikaciju, firestore baze podataka i sustava za pohranu datoteka. U liniji 7 nalaze se podaci o samom firebase projektu kako bi aplikacija prepoznala s kojim se točno projektom mora spojiti. U liniji 19 nalaze se metode koje će se koristiti u „SmartShop“ web aplikaciji na globalnoj razini.

```

22   boot: [
23     'i18n',
24     'axios',
25     'smartShopSetup'
26   ],

```

Slika 46. *quasar.conf.js* - *boot*

```
JS smartShopSetup.js X
smartshop > src > boot > JS smartShopSetup.js > ...
1  import firebase from 'firebase/app'
2  import 'firebase/auth'
3  import 'firebase/firestore'
4  import 'firebase/storage'
5
6  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
7  const firebaseConfig = {
8    apiKey: 'AIzaSyA... ',
9    authDomain: 'smartshop-... ',
10   projectId: 'smartshop-... ',
11   storageBucket: 'smartshop-... ',
12   messagingSenderId: '...',
13   appId: '1:...',
14   measurementId: 'G-...'
15 }
16
17 firebase.initializeApp(firebaseConfig)
18
19 export default ({ Vue }) => {
20   Vue.prototype.$auth = firebase.auth()
21   Vue.prototype.$db = firebase.firestore()
22   Vue.prototype.$storage = firebase.storage()
23 }
```

Slika 47. Programski kod „smartShopSetup.js“ datoteke

3.3.2. Router

U mapi `src/router` nalaze se dvije datoteke. U datoteci `index.js` nalazi se standardni kod koji se automatski generira prilikom izrade quasar aplikacije (Slika 48.). Ovdje se također provjerava ima li korisnik dopuštenje otići na zaštićene rute. U liniji 29 nalazi se metoda koja se poziva svaki put prije promjene rute. Prvo se provjerava je li za rutu potreban autorizirani pristup. Ako je potreban, onda se poziva metoda `isUserLoggedIn` koja se nalazi u liniji 7, koja provjerava je li korisnik ulogiran. Ova metoda vraća ili podatke o prijavljenom korisniku ili `null` vrijednost što znači da korisnik nije ulogiran. U slučaju da je korisnik ulogiran, dozvoljava mu se odlazak na željenu rutu, a ukoliko nije ulogiran onda ga se šalje na stranicu za login. Ukoliko nije potreban autorizirani pristup, korisniku se dopušta odlazak na željenu rutu.

Unutar datoteke `routes.js` definiraju se sve rute i komponente koje se učitavaju u aplikaciji (Slika 49.). Za rute od linije 2 do 18 nije potreban autorizirani pristup, odnosno korisnik ne mora biti prijavljen. Za rute od 19 do 28 linije korisnik mora biti ulogiran inače mu pristup neće biti dopušten. Ukoliko se ukuca neka nepostojeća ruta učitati će se stranica greške što je definirano u linijama 32 do 35.

```
JS index.js X JS routes.js
smartshop > src > router > JS index.js > ...
1 import Vue from 'vue'
2 import VueRouter from 'vue-router'
3 import routes from './routes'
4
5 Vue.use(VueRouter)
6
7 const isUserLoggedIn = () => {
8   return new Promise((resolve, reject) => {
9     const unsubscribeOnAuthStateChanged =
10     Vue.prototype.$auth.onAuthStateChanged(theUser => {
11       resolve(theUser)
12       unsubscribeOnAuthStateChanged()
13     }, err => {
14       console.error(err)
15       resolve(null)
16       unsubscribeOnAuthStateChanged()
17     })
18   })
19 }
20
21 export default function () {
22   const Router = new VueRouter({
23     scrollBehavior: () => ({ x: 0, y: 0 }),
24     routes,
25     mode: process.env.VUE_ROUTER_MODE,
26     base: process.env.VUE_ROUTER_BASE
27   })
28
29   Router.beforeEach(async (to, from, next) => {
30     if (to.matched.some(record => record.meta.auth)) {
31       await isUserLoggedIn()
32         .then(res => {
33           if (res) {
34             next()
35           } else {
36             next('/Login')
37           }
38         })
39     } else {
40       next()
41     }
42   })
43
44   return Router
45 }
```

Slika 48. index.js datoteka u mapi src/router

```
JS index.js JS routes.js X
smartshop > src > router > JS routes.js > ...
1  const routes = [
2    {
3      path: '/',
4      component: () => import('layouts/MainLayout.vue'),
5      children: [
6        { path: '', component: () => import('pages/Index.vue') },
7        { path: '/Phones', component: () => import('pages/Phones.vue') },
8        { path: '/SmartWatches', component: () => import('pages/SmartWatches.vue') },
9        { path: '/Accessories', component: () => import('pages/Accessories.vue') },
10       { path: '/Login', component: () => import('pages/Login.vue') },
11       { path: '/Registration', component: () => import('pages/Registration.vue') },
12       { path: '/ShoppingCart', component: () => import('pages/ShoppingCart.vue') },
13       { path: '/Delivery', component: () => import('pages/Delivery.vue') },
14       { path: '/Payment', component: () => import('pages/Payment.vue') },
15       { path: '/AboutUs', component: () => import('pages/AboutUs.vue') },
16       { path: '/Search', component: () => import('pages/Search.vue') }
17     ]
18   },
19   {
20     path: '/Account',
21     component: () => import('layouts/MainLayout.vue'),
22     meta: { auth: true },
23     children: [
24       { path: '/Account', meta: { auth: true }, component: () => import('pages/Account.vue') },
25       { path: '/Orders', meta: { auth: true }, component: () => import('pages/Orders.vue') },
26       { path: '/WishList', meta: { auth: true }, component: () => import('pages/WishList.vue') }
27     ]
28   },
29   // Always leave this as last one,
30   // but you can also remove it
31   {
32     path: '*',
33     component: () => import('pages/Error404.vue')
34   }
35 ]
36
37 export default routes
```

Slika 49. routes.js datoteka u mapi src/router

3.3.3. Prijava

Login.vue je komponenta koja služi za prijavu u sustav. Unutar template (Slika 50.) dijela nalazi se q-form komponenta koja služi za unos podataka za prijavu. Klikom na komponentu q-btn u liniji 24 pokreće se metoda *onLogin*. Unutar q-card komponente koja ima okvir (bordered) nalazi se više sekcija. Unutar prve sekcije nalazi se q-img koji predstavlja logo web trgovine širine 100px i naslov h4 odnosno naziv web trgovine. U slijedećoj sekciji nalaze se dva q-inputa. Prvi input služi za unos email adrese koji ima atribut *:rules* za provjeru ispravnosti oblika email adrese. Drugi input služi za unos lozinke. On je tipa *password* tako da prilikom unosa lozinke ona ostane skrivena, odnosno prikazuju se samo zvjezdice (*). Zatim slijedi q-card-actions komponenta unutar koje se nalazi q-btn koji je centriran, tipa *submit*, primarne boje, s nazivom 'Prijava' i ima uključene margine s lijeve i desne strane (*q-mx-lg*). Zatim slijedi komponenta q-separator primarne boje i s uključenom opcijom *inset*. Q-separator je zapravo samo horizontalna crta, a opcija *inset* znači da se crta na krajevima malo skрати. Ispod toga nalazi se nova sekcija sa informativnim textom. I na kraju imamo q-card-actions unutar kojeg se nalazi q-btn koji služi za promjenu rute. Pomoću atributa *'to'* prebacujemo se na stranicu za registraciju.

```

Login.vue X
smartshop > src > pages > Login.vue > {} "Login.vue" > script
1 <template>
2 <q-page>
3 <div class="q-pa-lg row justify-center">
4 <div class="col" style="max-width: 400px">
5 <q-form @submit="onLogin">
6 <q-card bordered>
7 <q-card-section>
8 <div class="flex flex-center">
9 <q-img width="100px" src="~/assets/smartshoplogo.png" />
10 <h4 class="q-ml-md">Smart<strong>Shop</strong></h4>
11 </div>
12 <div class="text-h5" align="center">Prijava</div>
13 </q-card-section>
14 <q-card-section>
15 <div class="q-gutter-sm">
16 <q-input filled type="email" v-model="email" label="Email adresa *" lazy-rules
17 :rules="[ val => emailPattern.test(val) || 'Molimo upišite valjanu email adresu' ]" />
18 <q-input filled type="password" v-model="password" label="Lozinka *" lazy-rules
19 :rules="[ val => val && val.length > 0 || 'Molimo upišite lozinku' ]" />
20 </div>
21 </q-card-section>
22 <div class="q-pb-sm">
23 <q-card-actions align="center">
24 <q-btn label="Prijava" type="submit" color="primary" class="q-px-lg" />
25 </q-card-actions>
26 </div>
27 <q-separator color="primary" inset />
28 <q-card-section align="center" >
29 <div class="text-h6">Novi korisnik? Registriraj se!</div>
30 </q-card-section>
31 <div class="q-pb-sm">
32 <q-card-actions align="center">
33 <q-btn to="/Registration" label="Registracija" color="primary" class="q-px-lg" />
34 </q-card-actions>
35 </div>
36 </q-card>
37 </q-form>
38 </div>
39 </q-page>
40 </template>
41

```

Slika 50. Template dio Login.vue komponente

Unutar script (Slika 51.) dijela nalazi se standardna vue metoda mounted koja se automatski pokreće kada je komponenta spremna. Unutar nje se provjerava dali je korisnik prijavljen te ako je, onda ga se automatski preusmjerava na novu rutu /Account. Zatim imamo metodu onLogin() koja se pokreće klikom na gumb *Prijava*. Unutar nje se poziva metoda za prijavu putem email adrese i lozinke. Ako je prijava uspješna izvodi se .then dio koji nas preusmjerava na rutu /Account. Ako prijava nije uspješna, izvodi se .catch dio koji će nas pomoću q.notify plugina obavijestiti o greški prilikom prijave.

```

43 <script>
44 export default {
45   name: 'LoginPage',
46   data () {
47     return {
48       emailPattern: /^(?=[a-zA-Z0-9@._%+-]{6,254}$)[a-zA-Z0-9._%+-]{1,64}@(?:[a-zA-Z0-9-]{1,63}\.){1,8}[a-zA-Z]{2,63}$/ ,
49       email: null,
50       password: null
51     }
52   },
53   mounted: function () {
54     if (this.$auth.currentUser) {
55       this.$router.push('/Account')
56     }
57   },
58   methods: {
59     onLogin () {
60       this.$auth.signInWithEmailAndPassword(this.email, this.password)
61         .then(response => {
62           this.$router.push('/Account')
63         })
64         .catch(error => {
65           const errorCode = error.code
66           if (errorCode === 'auth/wrong-password' || errorCode === 'auth/user-not-found') {
67             this.$q.notify({
68               type: 'negative',
69               message: 'Unijeli ste netočan email ili lozinku.'
70             })
71           } else {
72             this.$q.notify({
73               type: 'negative',
74               message: 'Prijava nije uspjela.'
75             })
76           }
77         })
78     }
79   }
80 }
81 </script>

```

Slika 51. Script dio Login.vue komponente

3.3.4. Registracija

Registration.vue komponenta služi za registraciju novih korisnika. Ova komponenta je jako slična Login.vue komponenti stoga će biti opisani samo različiti dijelovi. Template dio je skoro u potpunosti isti kao i kod Login.vue komponente. Razlika je u tome što se ovdje nalaze četiri inputa koji služe za unos imena, prezimena, email adrese i lozinke.

U script (Slika 52.) dijelu nalazi se onRegister metoda. Klikom na gumb *Registracija* poziva se ova metoda koja kreira novog korisnika pomoću emaila i lozinke. Ukoliko je registracija uspješna izvodi se .then blok unutar kojeg se izvedu različite radnje. Prvo se korisnika preusmjerava na rutu /Account. Zatim u lokalnu konstantu userId spremamo trenutni UID novoregistriranog korisnika. U liniji 97 unutar kolekcije 'users' stvara se novi dokument čiji će UID biti jednak konstanti userId. Unutar ovog dokumenta spremaju se tri podataka o korisniku, a to su ime, prezime i email adresa. Ukoliko su se podaci uspješno spremili unutar baze podataka izbacuje se notifikacija o uspješnoj registraciji. Ukoliko je došlo do greške prilikom spremanja podataka u bazu, u konzoli se ispisuje greška. Ako se registracija nije uspješno izvršila, izvodi se .catch blok unutar kojeg se provjerava dali je email adresa već u upotrebi, te ako je prikazuje se notifikacija s odgovarajućom porukom. Ako je greška bilo kojeg drugog tipa onda će se ispisati općenita poruka da registracija nije uspjela.

```

79 <script>
80 export default {
81   name: 'RegistrationPage',
82   data () {
83     return {
84       name: null,
85       surname: null,
86       emailPattern: /^(?=[a-zA-Z0-9@._%+-]{6,254}$)[a-zA-Z0-9._%+-]{1,64}@(?:[a-zA-Z0-9-]{1,63}\.){1,8}[a-zA-Z]{2,63}$/ ,
87       email: null,
88       password: null
89     }
90   },
91   methods: {
92     onRegister () {
93       this.$auth.createUserWithEmailAndPassword(this.email, this.password)
94         .then(response => {
95           this.$router.push('/Account')
96           const userId = this.$auth.currentUser.uid
97           console.log(userId)
98           this.$db.collection('users').doc(userId).set({
99             name: this.name,
100            surname: this.surname,
101            email: this.email
102          })
103          .then(() => {
104            this.$q.notify({
105              type: 'positive',
106              position: 'top',
107              message: 'Uspješno ste se registrirali.'
108            })
109          })
110          .catch((error) => {
111            console.error('Error writing document: ', error)
112          })
113        })
114        .catch(error => {
115          const errorCode = error.code
116          if (errorCode === 'auth/email-already-in-use') {
117            this.$q.notify({
118              type: 'negative',
119              message: 'Korisnički račun s ovom email adresom već postoji.'
120            })
121          } else {
122            this.$q.notify({
123              type: 'negative',
124              message: 'Registracija nije uspješna. Molimo pokušajte ponovo ili kontaktirajte korisničku podršku!'
125            })
126          }
127        })
128      }
129    }
130  }
131 </script>

```

Slika 52. Script dio Registration.vue komponente

3.3.5. Početna stranica

Index.vue komponenta je početna stranica „SmartShop“ web aplikacije. Unutar template (prikazano na *Slici 53. i 54.*) dijela nalazi se komponenta *q-responsive* s atributom *ratio*. Pomoću nje smo komponentu *q-carousel* učinili responzivnom u omjeru 8 naprema 3. To znači da će slike koje se nalaze u karuselu uvijek zadržati isti omjer neovisno o širini ekrana. Zatim slijedi blok s tri *q-card* komponente. Svaka kartica sadrži komponentu *q-img* koje imaju atribut *@click* pomoću kojeg pokrećemo metodu *goToPage* ukoliko se klikne na sliku. Prve dvije kartice još sadrže sekciju sa više *img* komponenta, dok treća kartica sadrži komponentu *q-list* unutar koje se nalazi više *q-item* komponenti. U template dijelu se još nalaze i dvije *q-dialog* komponente koje se aktiviraju klikom na gumb (vidjeti liniju 12 i 18). Prva dialog komponenta služi za dodavanje nove slike u karusel, dok druga služi za brisanje već postojeće slike iz

karusela. Unutar prve q-dialog komponente nalazi se komponenta q-file koja služi za odabir slike s lokalnog računala. Pomoću gumba u liniji 117 odustajemo od dodavanja nove slike, dok pomoću gumba u liniji 18 poziva se metoda `addImage` za dodavanje slike. U tom trenutku se prikaže i q-circular-progress komponenta koja nam daje do znanja da je u tijeku učitavanje slike u bazu podataka.

```

▼ Index.vue X
smartshop > src > pages > ▼ Index.vue {} "Index.vue"
1 <template>
2 <q-page>
3 <!-- Main page carousel -->
4 <q-responsive :ratio="8/3">
5 <q-carousel animated swipeable navigation infinite arrows
6 v-model="slide" :autoplay="autoplay" control-color="black"
7 transition-prev="slide-right" transition-next="slide-left"
8 @mouseenter="autoplay = false" @mouseleave="autoplay = true">
9 <q-carousel-slide v-for="(image, index) in images" :key="image.UIDImage"
10 :name="index" :img-src="image.imageUrl">
11 <q-btn dense color="red-14" icon="delete" v-if="admin"
12 @click="openImageDeleteDialog(image.UIDImage, image.imageName, index)">
13 <q-tooltip content-class="bg-red-14">Obriši trenutnu sliku</q-tooltip>
14 </q-btn>
15 </q-carousel-slide>
16 <template v-slot:control>
17 <q-carousel-control position="top-right">
18 <q-btn dense color="red-14" icon="add" v-if="admin" @click="addImageDialog = true">
19 <q-tooltip content-class="bg-red-14">Dodaj novu sliku</q-tooltip>
20 </q-btn>
21 </q-carousel-control>
22 </template>
23 </q-carousel>
24 </q-responsive>
25 <!-- Main page card menu -->
26 <div class="row justify-center q-gutter-md q-pa-md">
27 <q-card class="category-card">
28 <q-img src="~/assets/iphone-410324_1920.jpg" alt="Mobitel" basic @click="goToPage('/Phones')">
29 <div class="absolute-bottom text-h6">Mobiteli</div>
30 </q-img>
31 <q-card-section align="center">
32 Pregledaj našu ponudu mobitela.
33 <div class="row justify-center items-center q-gutter-md q-pt-sm">
34 
35 
36 
37 
38 
39 
40 
41 
42 </div>
43 </q-card-section>
44 </q-card>
45 <q-card class="category-card">
46 <q-img src="~/assets/smart-watch-821557_1920.jpg" alt="Pametni sat" basic @click="goToPage('/SmartWatches')">
47 <div class="absolute-bottom text-h6">Pametni satovi</div>
48 </q-img>
49 <q-card-section align="center">
50 Pametni satovi za svaku ruku.
51 <div class="row justify-center items-center q-gutter-md q-pt-sm">
52 
53 
54 
55 
56 
57 
58 
59 </div>
60 </q-card-section>
61 </q-card>
62 <q-card class="category-card">
63 <q-img src="~/assets/two-pin-3509490_1920.jpg" alt="Mobitel i punjac" basic @click="goToPage('/Accessories')">
64 <div class="absolute-bottom text-h6">Dodaci</div>
65 </q-img>
66 <q-card-section>
67 <q-list dense class="q-gutter-xs">
68 <q-item clickable :to="{ path: '/Accessories', query: { brand: 'Punjači i adapteri' }}">
69 <q-item-section avatar>

```

Slika 53. Template dio index.vue komponente - 1. dio


```

69     <q-item-section avatar>
70       <q-icon size="md" color="primary" name="arrow_right" />
71     </q-item-section>
72     <q-item-section>Punjači i adapteri</q-item-section>
73   </q-item>
74   <q-item clickable :to="{ path: '/Accessories', query: { brand: 'Kabeli' }}">
75     <q-item-section avatar>
76       <q-icon size="md" color="primary" name="arrow_right" />
77     </q-item-section>
78     <q-item-section>Kabeli</q-item-section>
79   </q-item>
80   <q-item clickable :to="{ path: '/Accessories', query: { brand: 'Zaštitne maskice i futrole' }}">
81     <q-item-section avatar>
82       <q-icon size="md" color="primary" name="arrow_right" />
83     </q-item-section>
84     <q-item-section>Zaštitne maskice i futrole</q-item-section>
85   </q-item>
86   <q-item clickable :to="{ path: '/Accessories', query: { brand: 'Zaštitne folije i stakla' }}">
87     <q-item-section avatar>
88       <q-icon size="md" color="primary" name="arrow_right" />
89     </q-item-section>
90     <q-item-section>Zaštitna folije i stakla</q-item-section>
91   </q-item>
92 </q-list>
93 </q-card-section>
94 </q-card>
95 </div>
96 <!-- Dialog for adding image to carousel -->
97 <q-dialog v-model="addImageDialog" persistent>
98   <q-card style="min-width: 350px">
99     <q-form @submit="addImage">
100       <q-card-section>
101         <div class="text-h6">Odabir slike</div>
102       </q-card-section>
103       <q-card-section class="q-py-none">
104         <q-file clearable filled v-model="image"
105           label="Odabir slike" accept="image/*"
106           :rules="[ val => val || 'Obavezno polje']"
107           hint="Za najbolji prikaz koristite sliku omjera 8:3"
108         >
109           <template v-slot:prepend>
110             <q-icon name="cloud_upload" />
111           </template>
112         </q-file>
113       </q-card-section>
114       <q-card-actions align="right" class="text-primary q-py-none">
115         <q-circular-progress :indeterminate="upload" size="md" color="blue" class="q-ma-md" />
116       <q-space />
117       <q-btn no-caps color="primary" label="Odustani" v-close-popup />
118       <q-btn no-caps color="primary" label="Dodaj sliku" type="submit" />
119     </q-card-actions>
120   </q-form>
121 </q-card>
122 </q-dialog>
123 <!-- Dialog for image delete -->
124 <q-dialog v-model="imageDeleteDialog" persistent>
125   <q-card>
126     <q-card-section align="center" class="flex flex-center">
127       <q-icon name="help_outline" size="xl" />
128       <span class="q-ml-sm text-subtitle1">Jeste li sigurni da želite obrisati ovu sliku?</span>
129     </q-card-section>
130     <q-card-actions align="right">
131       <q-btn no-caps label="Odustani" color="primary" v-close-popup />
132       <q-btn no-caps label="Obriši" color="primary" @click="deleteImage" />
133     </q-card-actions>
134   </q-card>
135 </q-dialog>
136 </q-page>
137 </template>

```

Slika 54. Template dio index.vue komponente - 2. dio

4. Zaključak

„SmartShop“ web aplikacija vrlo je jednostavna i intuitivna za korištenje. Izrađena je po svim pravilima i preporukama dobrog web dizajna. Navigacija je lako uočljiva te jednostavna za korištenje bez obzira na kojem se mjestu web aplikacije nalazimo. Kroz cijelu web aplikaciju korištene su tri osnovne boje: crna, siva i bijela te tri boje za naglašavanje bitnih elemenata, a to su: crvena, zelena i plava.

Iz perspektive kupca, aplikacija je korisna jer na brz i jednostavan način može pronaći proizvode koji ga zanimaju, proučiti karakteristike proizvoda te ga naposljetku i kupiti i to sve iz udobnosti vlastitoga doma.

S druge strane, iz perspektive prodavatelja, aplikacija je korisna jer lako može dodavati nove ili uređivati stare proizvode, te obrisati proizvode koje više nema u ponudi.

Radi povećanog korištenja web i mobilnih aplikacija uzrokovanog epidemijom, ali i općenito rastom i razvojem tehnologija ova aplikacija ima veliki potencijal kako za prodavatelje mobilnih uređaja tako i za kupce.

5. Literatura

1. „Web aplikacije - prednosti i nedostaci“, Blog, Horizont Solutions, pristupljeno 1.9.2021.
<https://www.horizont.com.hr/web-aplikacije---prednosti-i-nedostatci-3-blog>
2. Kristina Matic, „Uslijed pandemije, online kupovina se povećala za 15,5 %“, Blog, eCommerce Hrvatska, objavljeno 30. 11. 2020.
<https://ecommerce.hr/uslijed-pandemije-online-kupovina-se-povecala-za-155/>
3. Christensson Per, "Framework Definition", TechTerms, zadnje ažurirano 7.3.2013.
<https://techterms.com/definition/framework>
4. Miguel López Mamani, „What is Vue.js?“, Blog, Avantica, objavljeno 26.11.2019.
<https://www.avantica.com/blog/what-is-vue.js-and-how-do-we-use-it>
5. James Denman, „Node.js definition“, WhatIs.com, TechTarget, zadnje ažurirano travanj 2012., <https://whatis.techtarget.com/definition/Nodejs>
6. „What is database?“, Oracle, Database definition, pristupljeno 1.9.2021.
<https://www.oracle.com/database/what-is-database/>
7. „Cloud Storage“, Google, Firebase, zadnje ažurirano 8.9.2021.
<https://firebase.google.com/docs/storage>
8. „Cloud Firestore“, Google, Firebase, zadnje ažurirano 9.9.2021.
<https://firebase.google.com/docs/firestore>
9. Vangie Beal, „What is a Server?“, Webopedia, Server definition, zadnje ažurirano 28.5.2021., <https://www.webopedia.com/definitions/server/>
10. „Firebase documentation“, Google, Firebase, <https://firebase.google.com/docs>
11. „Vue.js framework“, Vue.js, <https://vuejs.org/>
12. „Quasar framework“, Quasar, <https://v1.quasar.dev/>
13. „000webhost“, 000webhost, <https://hr.000webhost.com/>
14. „HTML, CSS, JavaScript“, w3schools, <https://www.w3schools.com/>

6. Popis priloga

Poveznica na „SmartShop“ web aplikaciju: <https://smartshop2021.000webhostapp.com/#/>

7. Popis slika

| | |
|--|----|
| Slika 1. Izgled početne stranice VS Code-a | 6 |
| Slika 2. Primjer Frontend vs. Backend - Autor: Seobility - License: CC BY-SA 4.0 | 7 |
| Slika 3. Primjer HTML elementa | 8 |
| Slika 4. Primjer Inline CSS-a | 8 |
| Slika 5. Primjer Internal CSS-a | 9 |
| Slika 6. Način uključivanja External CSS-a | 9 |
| Slika 7. Cloud Storage "SmartShop" web aplikacije | 11 |
| Slika 8. Organizacija spremanja fotografija proizvoda | 11 |
| Slika 9. Baza podataka "SmartShop" web aplikacije | 12 |
| Slika 10. Primjer Web servera - Autor: vitteshagupta – License: CC BY-SA 2.0 | 13 |
| Slika 11. Cpanel 000webhost-a | 14 |
| Slika 12. Header "SmartShop" aplikacije | 14 |
| Slika 13. Polje za unos pretraživanja | 15 |
| Slika 14. Padajući gumb korisnika | 15 |
| Slika 15. Horizontalna navigacija sa padajućim izbornicima | 15 |
| Slika 16. Header na mobilnim uređajima | 16 |
| Slika 17. Polje za pretraživanje na mobilnim uređajima / Slika 18. Mobilna navigacija | 16 |
| Slika 19. Footer "SmartShop" web aplikacije | 17 |
| Slika 20. Početna stranica | 18 |
| Slika 21. Dijalozi za brisanje trenutne i dodavanje nove slike | 18 |
| Slika 22. Tooltip | 19 |
| Slika 23. Forma za prijavu / Slika 24. Forma za registraciju | 19 |
| Slika 25. Notifikacija o greški prilikom prijave / Slika 26. Notifikacija o greški prilikom registracije | 19 |
| Slika 27. Stranica "Mobiteli" | 21 |
| Slika 28. Dijaloški okvir za prikaz detaljnih informacija o proizvodu | 21 |
| Slika 29. Stranica "Mobiteli" ukoliko je prijavljen korisnik administrator | 22 |
| Slika 30. Dijaloški okvir za dodavanje novog proizvoda | 23 |
| Slika 31. Dijaloški okvir za uređivanje proizvoda | 23 |
| Slika 32. Stranica "Košarica" | 24 |
| Slika 33. Drugi korak: Dostava | 25 |
| Slika 34. Potvrda plaćanja | 25 |
| Slika 35. Treći korak: Plaćanje | 25 |
| Slika 36. Mobilni prikaz stranice "Košarica" | 26 |
| Slika 37. Stranica "Moj račun" | 27 |
| Slika 38. Stranica "Lista želja" | 28 |
| Slika 39. Stranica "Narudžbe" | 28 |
| Slika 40. Detalji narudžbe | 29 |
| Slika 41. Mobilni prikaz stranice "Narudžbe" / Slika 42. Mobilni prikaz Detalja narudžbe | 29 |
| Slika 43. Stranica "O nama" | 30 |
| Slika 44. Stranica "Dostava" | 30 |
| Slika 45. Stranica "Plaćanje" | 31 |
| Slika 46. quasar.conf.js - boot | 31 |
| Slika 47. Programski kod „smartShopSetup.js“ datoteke | 32 |

| | |
|---|----|
| Slika 48. index.js datoteka u mapi src/router | 33 |
| Slika 49. routes.js datoteka u mapi src/router | 34 |
| Slika 50. Template dio Login.vue komponente | 35 |
| Slika 51. Script dio Login.vue komponente | 36 |
| Slika 52. Script dio Registration.vue komponente..... | 37 |
| Slika 53. Template dio index.vue komponente - 1. dio..... | 38 |
| Slika 54. Template dio index.vue komponente - 2. dio..... | 39 |