

Web trgovina uz naplatu putem opće uplatnice

Ostović, Mislav

Undergraduate thesis / Završni rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:831238>

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

Download date / Datum preuzimanja: **2025-02-27**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Stručni prijediplomski studij Računarstvo/Elektrotehnika

**WEB TRGOVINA UZ NAPLATU PUTEM OPĆE
UPLATNICE**

Završni rad

Mislav Ostović

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju****Ocjena završnog rada na stručnom prijediplomskom studiju**

Ime i prezime pristupnika:	Mislav Ostović
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	AR 4807,
JMBAG:	0165085164
Mentor:	Marina Peko, dipl. ing. el.
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	mr. sc. Željko Štanfel
Član Povjerenstva 1:	Marina Peko, dipl. ing. el.
Član Povjerenstva 2:	Robert Šojo, univ. mag. ing. comp.
Naslov završnog rada:	Web trgovina uz naplatu putem opće uplatnice
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Izraditi web trgovinu koja nudi naplatu putem opće uplatnice te komunicira s korisnikom putem e-maila o stanju računa i pošiljke. Korisnik može biti registriran prilikom kupovine i skupljati bodove/ostvariti popuste ili biti gost. Uloga prodavača na stranici omogućava editiranje i postavljanje novih proizvoda za prodaju te brisanje istih. Besplatna poštarina se nudi ako se kupi proizvode u određenom iznosu. Tema rezervirana za: Mislav Ostović
Datum ocjene pismenog dijela završnog rada od strane mentora:	29.11.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Vrlo dobar (4)
Datum obrane završnog rada:	21.02.2025.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	25.02.2025.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 25.02.2025.

Ime i prezime Pristupnika:

Mislav Ostović

Studij:

Stručni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

AR 4807,

Turnitin podudaranje [%]:

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Web trgovina uz naplatu putem opće uplatnice**

izrađen pod vodstvom mentora Marina Peko, dipl. ing. el.

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

SADRŽAJ	2
1. UVOD	1
1.1. Zadatak završnog rada	2
1.2. Izazovi i problemi.....	2
2. PREGLED POSTOJEĆIH RIJEŠENJA	3
2.1. Usporedba s drugim aplikacijama i postojeća rješenja	3
3. KORIŠTENE TEHNOLOGIJE.....	6
3.1. React.....	6
3.2. Firebase.....	6
3.3. Mogućnosti unapređenja.....	7
4. FUNKCIONALNOST APLIKACIJE	8
4.1. Autentifikacija.....	8
4.1.1. Implementacija autentifikacije u web trgovini	8
4.1.2. Proces registracije korisnika u web trgovini uključuje nekoliko ključnih koraka:	8
4.1.3. Proces prijave korisnika uključuje sljedeće korake:	10
4.2. Uvod u naručivanje narudžbe.....	11
4.2.1. Odabiranje proizvoda za narudžbu.	11
4.2.2. Upravljanje narudžbom i proizvodima u košarici.....	12
4.2.3. Unos osobnih podataka.....	14
4.2.4. Generiranje uplatnice i plaćanje	15
4.3. Slanje uplatnice elektroničkom poštom	17
4.3.1. Priprema za slanje email-a sa uplatnicom.....	17
4.3.2. Praćenje stanja narudžbe putem e-mail sustava.....	19
5. ZAKLJUČAK.....	21
6. LITERATURA	23
7. SAŽETAK.....	24
8. ABSTRACT	25
9. ŽIVOTOPIS.....	26

1. UVOD

U današnje digitalno doba, e-trgovina postaje sve značajniji način poslovanja, omogućujući poduzećima da dosegnu globalno tržište, a korisnicima omogućuju jednostavnu kupnju proizvoda i usluga iz udobnosti svojih domova.[1] Internet trgovine su evoluirale od jednostavnih online kataloga do kompleksnih platformi koje nude širok raspon funkcionalnosti, uključujući personalizaciju korisničkog iskustva, integraciju različitih načina plaćanja i napredne metode za praćenje zaliha i analizu podataka. Razvoj web aplikacija za e-trgovinu zahtijeva poznavanje suvremenih tehnologija i platformi koje mogu osigurati učinkovito, sigurno i skalabilno rješenje.

E-trgovina, ili elektronička trgovina, predstavlja prodaju i kupnju proizvoda ili usluga putem interneta. Ova industrija je u stalnom porastu, zahvaljujući rastućoj popularnosti internetskog trgovanja, širokoj dostupnosti interneta i promjenama u potrošačkim navikama.[2] Ključne komponente uspješne web trgovine uključuju jednostavno i intuitivno korisničko sučelje, sigurnost transakcija, učinkovitu obradu narudžbi te integraciju s raznim sustavima plaćanja. U modernoj e-trgovini, uspješna implementacija ovih komponenti može rezultirati većim povjerenjem kupaca, višim stopama konverzije i dugoročnim poslovnim uspjehom.

Jedna od popularnih tehnologija za izgradnju korisničkih sučelja je *React*, JavaScript biblioteka koja omogućuje izgradnju dinamičnih i interaktivnih web aplikacija.[3] *React* se ističe svojom jednostavnošću, modularnošću i podrškom za ponovno korištenje komponenti, što omogućuje programerima brži razvoj i održavanje kompleksnih sustava. S druge strane, *Firebase*, kao platforma u vlasništvu Googlea, nudi moćan skup alata za *backend* podršku, uključujući bazu podataka u stvarnom vremenu, autentifikaciju korisnika, *hosting* i bez serverske funkcionalnosti. Ova kombinacija tehnologija omogućuje izgradnju aplikacija koje su ne samo funkcionalne, već i skalabilne te otporne na velike fluktuacije u prometu.

Ovaj rad se fokusira na razvoj web trgovine s naglaskom na integraciju sustava za generiranje i obradu uplatnica putem email sustava, koristeći *React* za *frontend* i *Firebase* za *backend*. Detaljno će se analizirati prednosti ovih tehnologija u kontekstu izgradnje web trgovine koja cilja na korisnike u regijama gdje je plaćanje uplatnicom još uvijek uobičajeno, kao što su zemlje Balkana i dijelovi Istočne Europe. Rad će prikazati detaljnu metodologiju izrade web trgovine, uključujući arhitekturu sustava, proces autentifikacije korisnika i upravljanje bazom podataka, generiranje uplatnica i praćenje pošiljke putem email sustava. Očekivani rezultati ovog projekta su

funkcionalna web aplikacija koja omogućuje korisnicima jednostavnu registraciju, pregled proizvoda, dodavanje proizvoda u košaricu, generiranje uplatnica za plaćanje te praćenje narudžbi.

Na kraju, bit će diskutirani potencijalni izazovi u razvoju ovakvog sustava, kao i moguća unaprjeđenja i dodatne funkcionalnosti koje bi mogle biti implementirane u budućnosti. Zaključak će obuhvatiti pregled postignutih rezultata, implikacije za budući razvoj e-trgovine te smjer u kojem ide online trgovina.

1.1. Zadatak završnog rada

Cilj ovog rada je razvoj web trgovine s integriranom funkcionalnošću za generiranje i obradu uplatnica, koristeći *React* za frontend i *Firebase* za backend. Ovakav pristup omogućuje brzu i učinkovitu izradu web aplikacije koja može odgovoriti na potrebe malih i srednjih poduzeća u pogledu online prodaje. Kroz implementaciju ove aplikacije, cilj je demonstrirati kako suvremene web tehnologije mogu poboljšati korisničko iskustvo i povećati efikasnost poslovanja, čak i u regijama gdje su tradicionalni oblici plaćanja još uvijek dominantni.

1.2. Izazovi i problemi

Tijekom razvoja web trgovine suočili smo se s nekoliko izazova, uključujući optimizaciju performansi aplikacije i osiguranje sigurnosti podataka korisnika. Korištenje *Reacta* i *Firebasea* pokazalo se kao učinkovit pristup za rješavanje tih problema.

2. PREGLED POSTOJEĆIH RIJEŠENJA

Web trgovine su prošle dug put od jednostavnih online kataloga do sofisticiranih platformi koje podržavaju različite oblike plaćanja, upravljanje zalihama i analizu korisničkog ponašanja. Suvremene web trgovine koriste napredne tehnologije za personalizaciju korisničkog iskustva, uključujući preporučivanje proizvoda na temelju prethodnih kupnji i pregleda. Korištenje umjetne inteligencije i strojnog učenja omogućuje trgovcima da prilagode ponudu svakom pojedinom korisniku, čime se povećava konverzija i zadovoljstvo kupaca. U današnje vrijeme, putem web trgovine korisnici mogu naručiti jako veliki raspon proizvoda, uključujući proizvode iz gotovo svake grane proizvodnje, što otvara nove mogućnosti za poslovanje. [4]

2.1. Usporedba s drugim aplikacijama i postojeća rješenja

Iako velik dio zemalja zahtijeva načine plaćanja poput PayPal, CashApp, kreditnih ili debitnih kartica, web trgovine na prostoru Republike Hrvatske i dalje, uz te načine, koriste i uplatnicu kao jedan od oblika plaćanja. Jedna od takvih stranica je Kapitalac, trgovina koja se bavi prodajom opreme za pecanje, mamaca, kuka, štapova, šatora, obuće i odjeće. Odabirom na plaćanje uplatnicom, ona se šalje na e-poštu kupca kako je prikazano na slici 2.1.

E-mail: nikola.ostovic5@gmail.com

Datum 11.11.2024 Rok isporuke Dostava DPD
Valjanost 16.11.2024 Narudžbenica K100005 /11.11.2024 Odgovorna osoba IVAN MALTAR
Način plaćanja Transakcijski Račun

Ponuda br. 24-012-000956

Naziv	Količina	MJ	Cijena Bez PDV-a	Rabat %	Cijena - Rabat	PDV %	Vrijednost bez PDV	
Daiwa 18 Regal LT 3000D-C	1,00	KOM	62,72	0,00	62,72	25,00	62,72	
Savage Gear Sg2 Fast Game 8ft 11" 2.71M F 15-50G Mmh 2Sec	1,00	KOM	57,00	10,00	51,30	25,00	57,00	
Daiwa J-Braid X8 0.22mm 150m Dark Green	1,00	KOM	14,80	10,00	13,32	25,00	14,80	
Por. br.							Ukupno	134,52
OIB							Popust	7,18
Slovima stopedesetdevet EUR 18/100							Vrijednost s popustom	127,34
						PDV	31,84	
						Za platiti EUR	159,18	
POREZNE STOPE	Osnova	PDV	Vrijednost					
Roba na koju se obračunava PDV po stopi 25%	127,34	31,84	159,18					

Molimo da uplatu izvršite na naš transakcijski račun pri PBZ broj:
HR25 2340 0091 1105 7518 6

Kod plaćanja gore navedenog iznosa navedite model **00** i poziv
24-012-000956.

Ovaj dokument je obrađen na računalu i vrijedi bez pečata i potpisa.
Reklamacija je moguća u roku 8 dana pismenim putem.

Ovaj dokument nije osnova za odbijanje pretporeza!



Slika 2.1. – Uplatnica sa web trgovine kapitalac.

U Hrvatskoj je plaćanje online putem uplatnice i dalje vrlo zastupljeno. Begen, online trgovina specijalizirana za prodaju knjiga, je još jedna od onih koje nude takav način plaćanja kako je prikazano na slici 2.2.

NALOG ZA NACIONALNA PLAĆANJA	
PLATITELJ (naziv/ime i adresa): Mislav Ostović Ulica Svetog Brtola, 70 35252 Sibinj	Hitno: <input type="checkbox"/> Valuta plaćanja: EUR Iznos: =2349 IBAN ili broj računa platitelja: Model: Poziv na broj platitelja: IBAN ili broj računa primatelja: HR1223600001101433117 Model: Poziv na broj primatelja: 163524 Šifra namjene: Opis plaćanja: Narudžba br 163524 Datum izvršenja:
PRIMATELJ (naziv/ime i adresa): BEGEN d.o.o. Milana Šenoae 18/0, Zagreb 10020 Zagreb	Model: Poziv na broj primatelja: 163524 Šifra namjene: Opis plaćanja: Narudžba br 163524 Datum izvršenja:
Obr. HUB SA 	Pečat korisnika PU Potpis korisnika PU
	Valuta i iznos: EUR 23,49 IBAN (račun) platitelja ili Platitelj: Model i poziv na broj platitelja: IBAN (račun) primatelja: HR1223600001101433117 Model i poziv na broj primatelja: 163524 Opis plaćanja: Narudžba br 163524 Ovjera:

Slika 2.2. - Begen web trgovine uplatnica

Begen, kao i Stivens, čija je uplatnica prikazana na slici 2.3., dokaz su da je plaćanje uplatnicom i dalje prisutno za sve vrste robe. Dok je Kapitalac za ribolov, Begen za knjige, tako je i Stivens za svakodnevnu odjeću i obuću.

UPLATNICU POPUNITE OVAKO

SLIKAJ I PLATI

UNIVERZALNI NALOG ZA PLAĆANJE		Prijavite se u vašu mobilnu aplikaciju za bankarstvo.
PLATITELJ (naziv/ime i adresa): Mislav Ostovic Ulica Svetog Brtola, 70 35252 Sibinj	Hitno: <input type="checkbox"/> Valuta plaćanja: EUR Iznos: 31,41 IBAN ili broj računa platitelja: Model: Poziv na broj platitelja:	Odaberite opciju 'Slikaj i plati', te skenirajte ovaj kod. Provjerite i potvrdite podatke za plaćanje.
IBAN ili broj računa primatelja: HR5424020061500080097	Model: Poziv na broj primatelja: HR00 4331 Šifra namjene: Opis plaćanja: Narudžba: 4331 - 18.11.2024. Datum izvršavanja:	
PRIMATELJ (naziv/ime i adresa): Legend d.o.o. Prolaz Andrije Kačića Miošića 2 43000 Bjelovar	Datum izvršavanja:	

Slika 2.3. - Stivens uplatnica.

3. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju obrađuju se osnovni koncepti i teorije relevantni za izradu web trgovine s uplatnicom koristeći *React* i *Firebase*. *Firebase* je koristan alat za početnike jer ne zahtijeva napredno programiranje. On automatski kreira, briše i ažurira podatke. *React*, s druge strane, zahtijeva poznavanje HTML-a, CSS-a i dobro razumijevanje JavaScript programskog jezika. Koristan je za izradu interaktivnog i dinamičnog korisničkog sučelja. Web trgovine omogućuju korisnicima kupnju proizvoda i usluga putem interneta, što zahtijeva učinkovite metode za upravljanje korisničkim sučeljem, bazom podataka i sustavom plaćanja. Razumijevanje ovih tehnologija ključno je za stvaranje funkcionalne, skalabilne i sigurne web trgovine.

3.1. React

React je popularna JavaScript biblioteka za izgradnju korisničkih sučelja koju je razvio Facebook. Omogućuje stvaranje dinamičnih web aplikacija koje su brze, prilagodljive i lako skalabilne. Glavne prednosti *React-a* uključuju virtualni DOM (engl. *Document Object Model*), komponentnu arhitekturu i mogućnost ponovne upotrebe koda.

Virtualni DOM ključna je tehnologija koja omogućuje brzo i učinkovito ažuriranje korisničkog sučelja, jer *React* ažurira samo one dijelove stranice koji su se promijenili, čime se smanjuje broj operacija na stvarnom DOM-u. Komponentna arhitektura omogućuje razbijanje aplikacije na manje, samostalne dijelove koji se mogu ponovno koristiti, što olakšava razvoj i održavanje aplikacije. Osim toga, *React* podržava jednosmjerni tijek podataka, što olakšava praćenje stanja aplikacije i smanjuje mogućnost pogrešaka.

3.2. Firebase

Firebase je platforma u vlasništvu Googlea koja pruža niz alata za razvoj web i mobilnih aplikacija. *Firebase* uključuje bazu podataka u stvarnom vremenu, autentifikaciju korisnika, hosting, cloud funkcije i još mnoge druge servise koji omogućuju brz razvoj aplikacija bez potrebe za upravljanjem serverima.

Jedna od ključnih komponenti *Firebase-a* je njegova *Firebase Firestore* baza podataka u stvarnom vremenu, koja omogućuje sinkronizaciju podataka između klijentske aplikacije i baze podataka u

stvarnom vremenu. To je *NoSQL* baza podataka koja koristi dokumentno-orijentirani model podataka, što znači da se podaci pohranjuju u obliku dokumenata i kolekcija, a ne u tablicama s redovima i stupcima kao kod relacijskih baza. To znači da promjene koje napravi jedan korisnik mogu biti odmah vidljive svim drugim korisnicima, što je posebno korisno u aplikacijama koje zahtijevaju brzu i efikasnu razmjenu informacija. *Firebase Authentication* omogućuje sigurno upravljanje korisničkim računima, dok *Firebase Hosting* nudi brz i pouzdan način za hosting web aplikacija.

3.3. Mogućnosti unapređenja

Buduća unaprjeđenja uključuju dodavanje novih izgleda sučelja, kao što su hamburger izbornik, bolji raspored proizvoda u košarici, animacije prilikom pritiska gumbova te funkcionalnosti poput praćenja narudžbi i za neprijavljene korisnike.

4. FUNKCIONALNOST APLIKACIJE

Aplikacija ima dvije uloge: prodavača, koji je administrator, te kupca, koji može biti prijavljen s korisničkim računom ili kao gost. Prodavač može dodavati i uklanjati proizvode iz online trgovine, kao i obavještavati kupca o stanju njegove narudžbe. Kupac koji se prijavi s korisničkim računom može skupljati bodove pri kupovini narudžbi kako bi ostvario dodatne popuste, kao i pratiti narudžbu putem sustava elektroničke pošte. Prikazana je i opisana funkcionalnost zadatka završnog rada putem prikaza pojedinih dijelova aplikacije i kodova koji ih obuhvaćaju.

4.1. Autentifikacija

U suvremenim web aplikacijama, posebno onima koje uključuju *e-commerce* funkcionalnosti, autentifikacija korisnika igra ključnu ulogu u osiguravanju sigurnosti i zaštiti osjetljivih podataka. Autentifikacija je proces kojim se provjerava identitet korisnika prije nego što im se omogući pristup određenim funkcionalnostima ili resursima aplikacije. Za potrebe ovog projekta, razvijenog pomoću *Reacta* i *Firebase-a*, implementirana je autentifikacija putem e-mail adrese i lozinke.

4.1.1. Implementacija autentifikacije u web trgovini

U implementaciji autentifikacije korišten je *Firebase Authentication*, usluga koja omogućuje jednostavnu integraciju različitih metoda autentifikacije, uključujući prijavu putem e-maila i lozinke, prijavu putem društvenih mreža, te druge napredne metode kao što su autentifikacija putem telefonskog broja ili dvofaktorska autentifikacija. *Firebase Authentication* odabran je zbog svoje jednostavnosti implementacije, visoke razine sigurnosti i podrške za više platformi, što omogućava dosljedno iskustvo korisnicima.

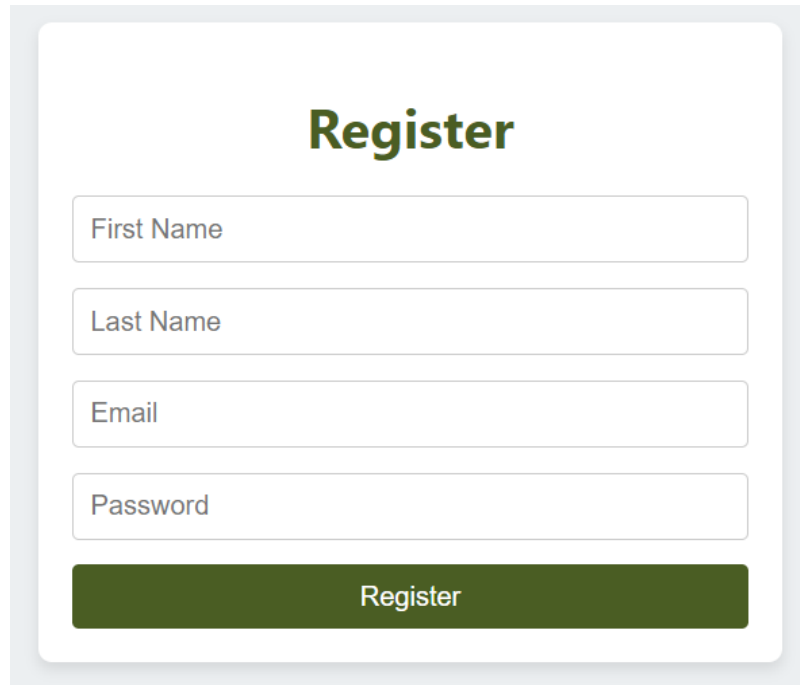
4.1.2. Proces registracije korisnika u web trgovini uključuje nekoliko ključnih koraka:

1. Unos korisničkih podataka: Kupac unosi osnovne podatke poput imena, prezimena, e-mail adrese i zaporke putem registracijskog obrasca. Ovi podaci zatim se šalju na *Firebase Authentication* server radi kreiranja novog korisničkog računa. Na slici 3.1. prikazan je uvoz funkcije zadužene za uvoz biblioteke za autentifikaciju.

```
import { getAuth, createUserWithEmailAndPassword, updateProfile } from 'firebase/auth';
```

Slika 3.1. - `createUserWithEmailAndPassword` iz Firebase biblioteke za autentifikaciju

Slika 3.2. prikazuje sučelje za registraciju korisnika koji želi napraviti korisnički račun.



The image shows a registration form with the title "Register" in a bold, green font. Below the title are four input fields: "First Name", "Last Name", "Email", and "Password". At the bottom of the form is a green button labeled "Register". The form is centered on a light gray background.

Slika 3.2. Izgled sučelja za registraciju

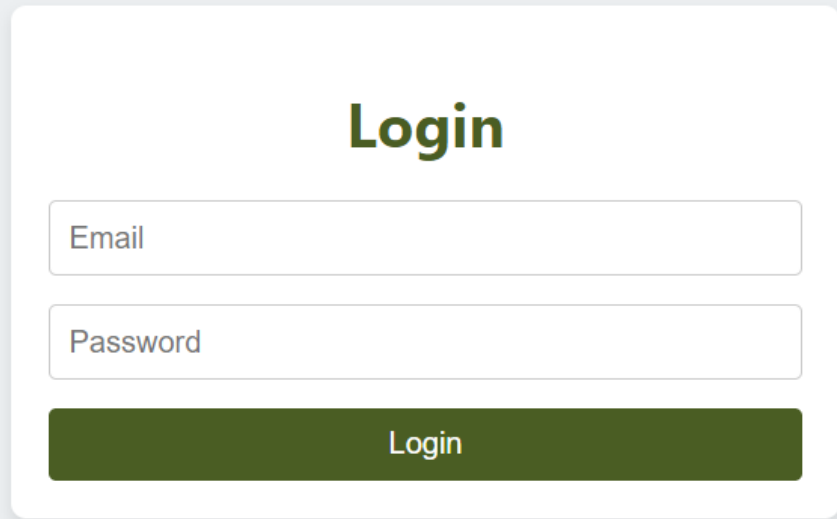
2. Kreiranje korisničkog računa: Funkcija `createUserWithEmailAndPassword` iz *Firebase Authentication* biblioteke koristi se za kreiranje novog korisničkog računa. Ako je kreiranje računa uspješno, korisnički se podaci spremaju u *Firestore*, bazu podataka u oblaku koja je također dio *Firebasea*. Ovdje se pohranjuju dodatni podaci o korisniku, kao što su ime, prezime i početna uloga koja je *user*. Korisnik koji je administrator ima ulogu prodavača i mora biti ručno unesen odlaskom na *Firestore* bazu podataka, te mu uloga biva promijenjena iz *user* u *admin*.

3. Ažuriranje profila korisnika: Nakon što je korisnički račun kupca uspješno kreiran, koristeći funkciju `updateProfile`, aplikacija ažurira kupčev profil dodavanjem njihovog imena i prezimena. Ova funkcionalnost omogućuje personaliziranije korisničko iskustvo i jednostavnije upravljanje korisničkim računima.

4. Prijava i upravljanje sesijom: Nakon uspješne registracije, korisnik je automatski prijavljen pomoću funkcije `login` unutar *AuthContexta*, što omogućuje aplikaciji da upravlja sesijom korisnika i osigura pristup odgovarajućim resursima.

4.1.3. Proces prijave korisnika uključuje sljedeće korake:

1. Unos korisničkih podataka: Korisnik unosi svoju e-mail adresu i lozinku u prijavni obrazac. Nakon slanja obrasca, aplikacija koristi funkciju *signInWithEmailAndPassword* iz *Firestore Authentication* biblioteke za provjeru korisničkih podataka. Na slici 3.3. prikazan je izgled sučelja za prijavu korisnika.

The image shows a login interface within a light gray rounded rectangle. At the top center, the word "Login" is displayed in a bold, dark green font. Below this, there are two white input fields with thin gray borders. The first field is labeled "Email" and the second is labeled "Password". At the bottom of the form is a solid dark green rectangular button with the word "Login" written in white text in the center.

Slika 3.3. Izgled sučelja za prijavu registriranog korisnika

2. Provjera identiteta: Ako su korisnički podaci točni, korisniku se dopušta pristup web trgovini i upravljanje njihovom sesijom putem *AuthContexta*. U slučaju neuspješne prijave, korisnik dobiva odgovarajuću poruku o pogrešci.

3. Upravljanje pogreškama i povratnim informacijama: Za oba procesa, registraciju i prijavu, implementirano je detaljno upravljanje pogreškama i povratnim informacijama. Korisnici dobivaju jasne i informativne poruke o uspjehu ili neuspjehu svojih akcija, čime se poboljšava korisničko iskustvo i osigurava pravovremena povratna informacija.

U ovom projektu, implementacija autentifikacije putem *Firestore Authentication* pokazala se kao optimalan izbor zbog svoje jednostavnosti integracije i fleksibilnosti u podršci za različite platforme i metode prijave.[5] Korištenje emaila i lozinke kao primarne metode autentifikacije osigurava pristup korisnicima na intuitivan način, dok dodatne funkcionalnosti poput upravljanja korisničkim profilima i sesijama pružaju personalizirano i sigurno korisničko iskustvo. *Firestore Authentication* omogućuje brzu i pouzdanu provjeru identiteta korisnika te olakšava upravljanje različitim vrstama korisnika unutar sustava, uključujući osnovne korisnike i administratore. Kroz

ovaj pristup, web trgovina ne samo da osigurava sigurnost osjetljivih podataka, već i poboljšava cjelokupno korisničko iskustvo omogućujući jednostavan i siguran način prijave i registracije.

4.2. Uvod u naručivanje narudžbe

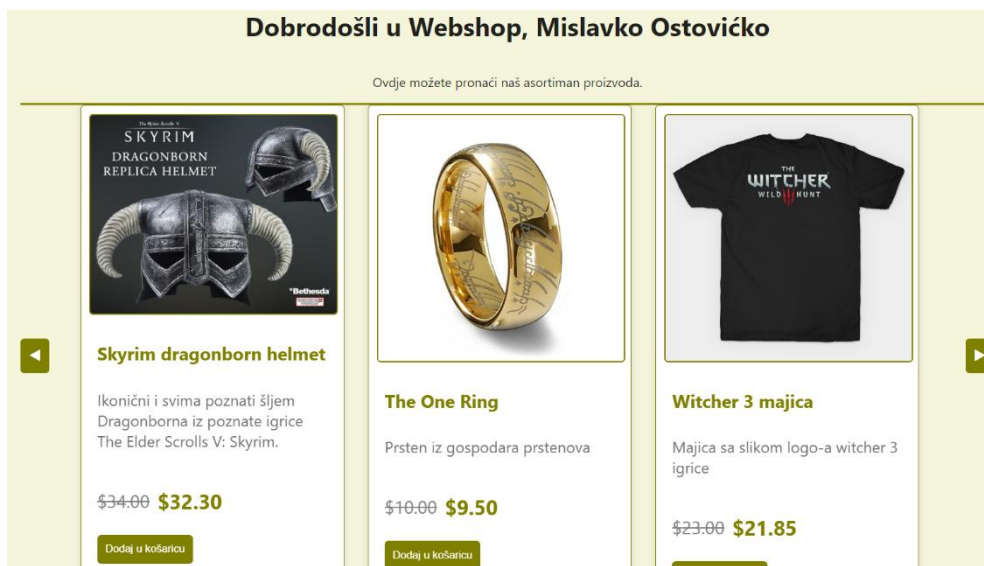
Naručivanje proizvoda putem predstavljene web trgovine omogućava jednostavno i intuitivno iskustvo kupovine. Nakon odabira željenih proizvoda u našoj sekciji Home, korisnik ima mogućnost dodavanja tih proizvoda u košaricu, gdje može upravljati njihovim količinama, iskoristiti bodove za popust i pregledati ukupne troškove narudžbe. Home.js komponenta služi kao početna stranica web trgovine, gdje korisnici mogu pregledavati dostupne proizvode i dodavati ih u svoju košaricu. Nakon što odaberu proizvode koje žele kupiti, Cart.js omogućava detaljnije upravljanje narudžbom, uključujući prilagodbu količina proizvoda, iskorištavanje bodova za popust i izračunavanje bodova koje će zaraditi ovom kupovinom. Nakon što budu zadovoljni izborom i detaljima svoje narudžbe, sljedeći korak je unos svojih osobnih podataka putem UserDetails.js komponente. Ovdje korisnik unosi svoje ime, prezime, adresu i ostale potrebne podatke kako bi se osigurala ispravna dostava i komunikacija vezana uz narudžbu.

Završni korak u procesu naručivanja je generiranje uplatnice i plaćanje putem Payment.js komponente. Ovdje korisnik može generirati PDF verziju svoje uplatnice, koja se potom šalje na njegovu email adresu samo u slučaju da je korisnik prijavljen. Nakon što je uplata uspješno izvršena, korisnikov račun se ažurira s novim bodovima lojalnosti, koji se mogu koristiti za buduće popuste.

Ovaj cjelokupni proces dizajniran je da bude što jednostavniji i transparentniji za korisnika, omogućujući brzu i sigurnu kupovinu s dodatnim prednostima kroz sustav bodova lojalnosti.

4.2.1. Odabiranje proizvoda za narudžbu.

Na početnoj stranici *Home.js*, klikom na gumb "Dodaj u košaricu", korisnik dodaje proizvod koji želi u košaricu, kao što je prikazano na slici 3.4.



Slika 3.4. - Početna stranica korisnika

Gumb „Dodaj u košaricu“ nalazi se u komponenti `Product.js`. Funkcija `onAddToCart` se prosljeđuje iz `Home.js` kao `prop` u `Product` komponentu. Klikom na „Dodaj u košaricu“ gumb, poziva se ista funkcija sa `onClick handler-om` i trenutnim proizvodom kao argumentom kako je objašnjeno na slici 3.5.

```
<button className="add-to-cart-button" onClick={() => onAddToCart(product)}>
  Dodaj u košaricu
</button>
```

Slika 3.5. - Gumb "Dodaj u košaricu iz `Product.js`

Prijavljeni korisnici također ostvaruju i 5 posto popusta na sve proizvode što se vidi i u `handleAddToCart` funkciji. Funkcija koristi `dispatch` iz `CartContext.js` tako što prima objekt akcije s tipom `ADD_TO_CART` i `payload` koji sadrže detalje o proizvodu, cijenu proizvoda te količinu koja je postavljena na jedan pri kliku za dodavanje. `Dispatch` funkcija koristi i reduktor za provjeravanje stanja proizvoda u košarici. Nakon svakog pritiska na gumb, stanje u košarici u navigacijskoj traci povećava se za jedan. Pritiskom na gumb košarice, komponenta 'Link' nas vodi prema `/cart` putanji definiranoj u `App.js` te sa sobom ima i `getTotalCartItems` funkciju koja nam prikazuje broj proizvoda u košarici.

4.2.2. Upravljanje narudžbom i proizvodima u košarici

Odlaskom na košaricu prikazanu na slici 3.6. mogu se vidjeti proizvodi koji su dodani putem naslovne stranice. Svakom se proizvodu može se dodatno povećati količina pritiskom na gumb + ili smanjiti količina gumbom -. Ukoliko se želi ukloniti proizvod u potpunosti, može se kliknuti na gumb ukloni iz košarice.

Košarica

The One Ring
Cijena: \$28.50
Količina: 3

-
+
Ukloni iz košarice

Skyrim dragonborn helmet
Cijena: \$64.60
Količina: 2

-
+
Ukloni iz košarice

Ukupno: \$93.10

Popust: \$9.31

Ukupno s popustom: \$83.79

Uložili ste 50 bodova i ostvarili 10% popusta

-
+

Nemate dovoljno bodova.

Ukupno s bodovima: \$83.79

Ovom narudžbom zaradit ćete 46 bodova.

Naruči

Slika 3.6. - Prikaz košarice registriranog korisnika

Ukupno prikazuje cijenu koju korisnik plaća sa 5 posto popusta zbog registracije i prijave. Ukoliko korisnik uloži 50 bodova, ostvarit će 10 posto popusta na ukupnu cijenu. Taj popust prikazan je ispod ukupne cijene. Korisnik može dodati najviše 250 bodova, što rezultira popustom od 50 posto. Ukoliko korisnik odluči dodati više bodova nego što ima, ispisat će se poruka „Nemate dovoljno bodova“ i korisnik neće moći ostvariti dodatan popust. Ukoliko, kao na slici 3.7., ima preko 250 bodova i želi još popusta, on neće moći dodati bodove zbog ograničenja od 50 posto, tj. 250 bodova.

Uložili ste 100 bodova i ostvarili 20% popusta

-
+

Nemate dovoljno bodova.

Ukupno s bodovima: \$80.56

Slika 3.7. - Nedovoljno bodova za ostvariti popust

Za svaka 2 dolara potrošena, korisnik ostvaruje 1 bod. To se vidi iz varijable *calculatePointsToEarn*

Varijabla provjerava je li korisnik prijavljen pomoću *currentUser* ? te računa koliko se ostvaruje bodova, u suprotnom, postavlja bodove na nula. Zatim se varijabla sprema u *setPointsToEarn*

funkciju koja se dobiva iz *useState hook-a* u *Reactu*. [6] Na kraju se ukupni ostvareni bodovi ispisuju pri dnu stranice. Pritiskom na gumb naruči, korisnik je odveden na stranicu za unos osobnih podataka.

4.2.3. Unos osobnih podataka

Slika 3.8. prikazuje stranicu za unos podataka, gdje korisnik unosi svoje ime, prezime, adresu, poštanski broj te kontakt telefon. Ovi podaci omogućuju administratorima da komuniciraju s korisnikom. Ime, prezime, adresa i poštanski broj bit će prosljeđeni na generiranu uplatnicu kasnije u procesu narudžbe.



Slika 3.8. - Prikaz sučelja za unos osobnih podataka

Podaci se prosljeđuju u *Payment.js*, komponentu za generiranje i slanje uplatnice, pomoću *navigate* funkcije iz *react-router-dom* biblioteke. Podaci pristignu iz košarice te zajedno sa podacima korisnika se šalju u komponentu za generiranje uplatnice kao na slici 3.9.

```
const handlePayment = () => {
  if (!orderData) {
    alert('Podaci o narudžbi nisu pronađeni.');
```

```
    return;
  }

  const completeOrderData = {
    ...orderData,
    ...userDetails,
  };

  navigate('/payment', { state: { completeOrderData } });
};
```

Slika 3.9. - *completeOrderData* koji šalje podatke prema generiranju uplatnice

4.2.4. Generiranje uplatnice i plaćanje

Nakon pritiska gumba na naplatu, korisnik je odveden na stranicu sa uplatnicom. Uplatnica podatke koje je primila, osim kontakt telefona, ispisuje u okviru informacija kupca. Isti okvir sadrži i 23 polja za unos IBAN-a. Svakim unesenim slovom ili znamenkom unos se prebacuje automatski na sljedeće polje sve dok korisnik ne unese cijeli IBAN. Opis plaćanja može, a i ne mora, biti unesen kako bi se uplatnica generirala. IBAN, naime, mora biti unesen kako bi se uplatnica mogla poslati. Popis proizvoda nalazi se u okviru pod nazivom proizvodi. Na dnu stranice nalazi se *QR* kod nastao generiranjem iz *qrcode.react* biblioteke koja je instalirana pomoću *npm install qrcode.react* komandom. [7] *QR* kod smješten je u *qr* kod kontejner u *div* klasi. Kako bi se uplatnica generirala i spremila kao pdf, poziva se *funkcija handleDownloadInvoice* nakon klika za slanje na email. Ta metoda poziva *createPdfBlob* funkciju koja generira pdf. Ista funkcija koristi *html2canvas* biblioteku koja mora biti dodatno instalirana.[8]Metoda *sendEmailWithAttachment* se poziva sa generiranim pdf *blob-om*, a ona koristi *emailjs* biblioteku o kojoj će kasnije biti pisano. Ista metoda *FormData* objekt u kojoj dodaje privitak i potrebne podatke, a zatim šalje *POST* zahtjev za *EmailJS API*. Kod zadužen za kreiranje uplatnice prikazan je na slici 3.10.

```
const createPdfBlob = async (input) => {
  const canvas = await html2canvas(input, { scale: 2 });
  const imgData = canvas.toDataURL('image/jpeg', 0.7);

  const pdf = new jsPDF('p', 'mm', 'a4');
  const pageWidth = pdf.internal.pageSize.getWidth();
  const pageHeight = pdf.internal.pageSize.getHeight();

  const margin = 15;
  const imgWidth = pageWidth - 2 * margin;
  const imgHeight = (canvas.height * imgWidth) / canvas.width;


  const backgroundCanvas = document.createElement('canvas');
  const context = backgroundCanvas.getContext('2d');
  backgroundCanvas.width = canvas.width + 2 * margin;
  backgroundCanvas.height = canvas.height + 2 * margin;
  context.fillStyle = 'white';
  context.fillRect(0, 0, backgroundCanvas.width, backgroundCanvas.height);
  context.drawImage(canvas, margin, margin);
  const imgDataWithBorder = backgroundCanvas.toDataURL('image/jpeg', 0.7);
  const pdfHeightWithBorder = imgHeight + 2 * margin;
  pdf.drawImage(imgDataWithBorder, 'JPEG', margin, margin, imgWidth, imgHeight);

  return pdf.output('blob');
```

Slika 3.10. - *createPdfBlob* funkcija za generiranje PDF-a

Pritiskom na gumb pošalji uplatnicu na email, iskače tast koji ispisuje da je narudžba poslana na mail. Također, pritiskom na gumb uplatnica se automatski šalje na email prijavljenog korisnika u pdf formatu te se narudžba sprema u bazu podataka. U slučaju da korisnik nije prijavljen, tj. da naručuje narudžbu kao gost, on neće imati uplatnicu poslanu na e-mail nego će se ona preuzeti na računalo korisnika u pdf formatu putem *downloadPdf* funkcije. Nakon toga, korisnik je preusmjeren na stranicu zahvale 2 sekunde nakon slanja narudžbe. U svojoj elektroničkoj pošti, korisnik izvršava plaćanje putem *qr* koda ili pomoću IBAN-a kao na slici 3.11.

Uplatnica

Primatelj <hr/> Mislav's webshop Izmišljena ulica, 123 98765, Nigdjezemska IBAN: HR123456789123456789123																																		
Platitelj <hr/> Mislav Ostovic Ulica Svetog Brtola 31000 IBAN platitelja: <table border="1"><tr><td>H</td><td>R</td><td>3</td><td>7</td><td>1</td><td>2</td><td>3</td><td>9</td><td>8</td><td>7</td><td>1</td><td>2</td><td>9</td><td>8</td><td>3</td><td>7</td><td>1</td></tr><tr><td>2</td><td>3</td><td>2</td><td>4</td><td>4</td><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	H	R	3	7	1	2	3	9	8	7	1	2	9	8	3	7	1	2	3	2	4	4	4											
H	R	3	7	1	2	3	9	8	7	1	2	9	8	3	7	1																		
2	3	2	4	4	4																													
Opis plaćanja: Plaćanje 3.9. 2024- Putem QR koda poslanog na mail mislav.ostovic82@gmail.com																																		
Proizvodi <hr/> Witcher 3 majica x 2 Star wars jedi ogrlica x 2 The One Ring x 3																																		
																																		
Pošalji uplatnicu na email																																		

Slika 3.11.- uplatnica koja će biti poslana na email

4.3. Slanje uplatnice elektroničkom poštom

Kako bi se omogućila automatska dostava uplatnice korisnicima, koristi se *EmailJS* usluga, koja omogućava jednostavno slanje e-mailova bez potrebe za vlastitim poslužiteljem.[9] *EmailJS* je vrlo koristan alat za web-aplikacije koje trebaju slati e-maile direktno iz *frontenda*, a jednostavno se integrira s JavaScript aplikacijama bez kompleksnih postavki.

U ovoj aplikaciji, *EmailJS* je iskorišten za slanje uplatnice u PDF formatu na e-mail adresu korisnika. Korisnik prvo ispunjava potrebne podatke u obrascu za plaćanje, uključujući vlastiti IBAN i opis plaćanja. Nakon što su podaci uneseni, generira se PDF dokument koji sadrži sve relevantne podatke, uključujući informacije o narudžbi i *QR* kod za plaćanje. PDF uplatnica se generira pomoću biblioteke *html2canvas* za pretvaranje HTML sadržaja u sliku, te *jsPDF* za pretvaranje te slike u PDF format.

Kada je PDF generiran, koristi se *EmailJS* za slanje e-maila korisniku. Ovaj proces uključuje slanje podataka kroz *EmailJS* API, gdje se koristi predložak e-maila koji smo prethodno definirali unutar platforme. U aplikaciji je integriran *EmailJS* pomoću njihovog API-ja, a potrebno je postaviti *service_id*, *template_id* i *user_id* što su identifikatori koje dobijemo prilikom registracije na *EmailJS* platformi. Na taj način PDF uplatnica se prilaže e-mailu i šalje na adresu korisnika koji je prijavljen u aplikaciju.

Glavna prednost korištenja *EmailJS*-a je ta što ne zahtijeva složene *backend* postavke za slanje e-mailova, čime se značajno olakšava implementacija ove funkcionalnosti u malim aplikacijama poput ovog web shopa.

4.3.1. Priprema za slanje email-a sa uplatnicom

U *Payment*, nakon kreiranja uplatnice sa podacima, slanje elektroničke pošte vrši se putem funkcije *sendEmailWithAttachment*. Prvi korak je uzimanje e-mail adrese korisnika koji je trenutno prijavljen. Nakon toga se u funkciji kreira *FormData* ako na slici 3.12. objekt u kojeg se pohranjuju svi potrebni podaci iz *EmailJS* API, uključujući ID servisa, ID predloška e-maila, ID korisnika *EmailJS*-a, adresu korisnika te PDF dokument kao prilog.

```

const formData = new FormData();
formData.append('service_id', 'service_6wfi8i5');
formData.append('template_id', 'template_h4kktuy');
formData.append('user_id', 'i_NoL9fQISBK9KZYH');
formData.append('user_email', userEmail);
formData.append('attachment', pdfBlob, 'uplatnica.pdf');

```

Slika 3.12. - *formData* objekt

Koristi se i *fetch* metoda da bi se poslao *POST* zahtjev na *EmailJS endpoint*, a šalje se PDF kao privitak u pošti. Ovisno o ishodu, korisniku se prikazuje poruka o uspješnom ili neuspješnom slanju uplatnice na e-mail. Nakon uspješnog slanja, korisniku stiže predložak na email u kojem korisnik mora dovršiti plaćanje kako bi se paket poslao. Predložak je prikazan na slici 3.13. Da bi predložak mogao slati uplatnice u pdf formatu, potrebno je plaćati mjesečnu pretplatu za uspješno slanje, u suprotnom, neće moći uplatnica biti poslana putem e-pošte.

The image shows a web-based email template editor. On the left, the 'Content' tab is active, displaying a draft email. The subject is 'Završite plaćanje'. The body text includes a greeting, a 48-hour deadline notice, and a bolded instruction: 'Nakon što platite, odgovorite na mail da ste platili.' Below this is a signature: 'Sve najbolje, Mislav's webshop.' On the right, the 'Settings' tab is active, showing fields for 'To Email' (with a placeholder {{user_email}}), 'From Name' (Mislav's webshop), 'From Email' (with a checked 'Use Default Email Address' option), 'Reply To' (with a placeholder {{reply_to}}), 'Bcc', and 'Cc'.

Slika 3.13. - *EmailJS template* za slanje uplatnice

Slika 3.14. prikazuje izgled automatske poruke poslana na elektroničku poštu.



Slika 3.14. - e-mail za završavanje plaćanja

Nakon plaćanja uplatnicom, bilo pomoću IBAN-a ili pomoću *qr* koda, korisnik odgovara na mail da je plaćeno. Zatim administrator web trgovine prima isti odgovor putem svoje e-pošte.

4.3.2. Praćenje stanja narudžbe putem e-mail sustava

Nakon što administrator provjeri je li uplata stvarno prošla, u svom *admin dashboardu* odlazi na uplaćenu narudžbu i označava je kao uplaćenu, kako prikazuje slika 3.15.. Nakon što je označio, automatski se šalje e-mail na adresu naručitelja s obavijesti o zaprimljenoj uplati.

Status narudžbe

- Uplata zaprimljena
- Paket predan dostavnoj službi
- Paket će biti dostavljen tokom dana
- Dostavljeno

Slika 3.15. - sučelje za status narudžbe koje *admin* označava

Na korisničkom sučelju prikazane različite faze u procesu narudžbe putem *checkbox*-ova. Postoje četiri različita stanja koja se mogu ažurirati. To su: uplata zaprimljena, paket predan dostavnoj službi, paket će biti dostavljen tijekom dana i dostavljeno. Svaki od ovih statusa vezan je za odgovarajući *checkbox*, a korisnik može ažurirati stanje klikom na bilo koji od njih. Svaki *checkbox* povezan je sa statusom narudžbe putem atributa *statuses*. Pritiskom na određenu kućicu poziva se funkcija *handleCheckboxChange(statusKey)*. Kada korisnik klikne na *checkbox*, status narudžbe se mijenja. To se postiže promjenom vrijednosti odgovarajućeg ključa u *statuses* objektu prikazanom na slici 3.16. Nakon što se promijeni status u sučelju, status narudžbe se ažurira i u *Firestore* bazi podataka pomoću funkcije *updateDoc()*. Ovo osigurava da promjena nije samo lokalna, već da se zapisuje u bazi. Nakon ažuriranja statusa u bazi podataka, aplikacija provjerava koji je status promijenjen, te se na temelju toga šalje odgovarajući email korisniku. Za ovo se koristi funkcija *sendStatusUpdateEmail(statusKey)*. Email se šalje putem *EmailJS* API-ja.

```
let templateId = '';  
  
switch (statusKey) {  
  case 'paymentReceived':  
    templateId = 'paymentRecieved';  
    break;  
  case 'handedOverToDelivery':  
    templateId = 'packageGivenForDelivery';  
    break;  
  case 'outForDelivery':  
    templateId = 'deliveryToday';  
    break;  
  case 'delivered':  
    templateId = 'template_kl0k9in';  
    break;  
  default:  
    console.error('Invalid statusKey:', statusKey);  
    return;  
}
```

Slika 3.16. - biranje odgovarajućeg *template id*-ja na osnovu statusa

Jednom kada se odredi koji obrazac treba koristiti, poziva se *EmailJS* API kako bi poslao email korisniku. To se postiže pomoću funkcije *emailjs.send()*, koja šalje zahtjev na *EmailJS* server s potrebnim podacima: *service_id*, *template_id*, *template_params*. Ako email bude uspješno poslan, ispisuje se poruka o uspjehu u konzolu. U slučaju greške, greška se također ispisuje u konzolu.

5. ZAKLJUČAK

U ovom završnom radu postavljen je zadatak za razvoj moderne web trgovine s funkcionalnošću integriranom za generiranje i obradu uplatnica za generiranje i obradu uplatnica, koristeći *React* za *frontend* i *Firebase* za *backend*. Svi ključni ciljevi postavljeni u zadatku završnog rada su ostvareni, a rezultati su prikazani i analizirani kroz nekoliko poglavlja.

U prvom dijelu rada objašnjeni su osnovni koncepti i teorijske podloge koje su relevantne za izradu web trgovine. Detaljno je opisana važnost e-trgovine u današnjem poslovnom okruženju te su predstavljene prednosti korištenja *Reacta* i *Firebasea*. *React* je odabran zbog svoje jednostavnosti i fleksibilnosti u izradi dinamičnih korisničkih sučelja, dok *Firebase* pruža robusne alate za *backend* podršku, uključujući bazu podataka, autentifikaciju i hosting.

U srednjem dijelu rada prikazano je konkretno rješenje zadatka. Razvijena je aplikacija koja korisnicima omogućuje jednostavnu registraciju, pregled proizvoda, dodavanje proizvoda u košaricu te generiranje uplatnica za plaćanje. Opisana je arhitektura sustava, uključujući način na koji *React* upravlja korisničkim sučeljem i kako *Firebase* rukuje bazom podataka i autentifikacijom korisnika. Generiranje uplatnica implementirano je tako da sustav stvara uplatnice s detaljima narudžbe i podacima koje je korisnici unesu, koje korisnici mogu preuzeti ili platiti putem *QR* koda pomoću mobilnog uređaja. Ono što ova aplikacija ima za razliku od ostalih web shopova jest komunikacija sa administratorom pomoću *messenger* sučelja, kao i komunikaciju putem email sustava.

Kroz rad su također identificirani izazovi i problemi s kojima smo se suočili tijekom razvoja aplikacije. To uključuje optimizaciju performansi i osiguranje sigurnosti podataka korisnika. Korištenje modernih web tehnologija kao što su *React* i *Firebase* pokazalo se kao učinkovit pristup za rješavanje ovih problema, no također su navedene mogućnosti za daljnje unaprjeđenje sustava. Predložena poboljšanja uključuju dodatne funkcionalnosti kao što su praćenje narudžbi u stvarnom vremenu za neprijavljene korisnike, integracija s drugim sustavima plaćanja te personalizacija korisničkog iskustva.

Glavni postignuti rezultat ovog rada je funkcionalna i skalabilna web trgovina koja može zadovoljiti potrebe malih i srednjih poduzeća u pogledu online prodaje. Aplikacija je dizajnirana tako da bude lako proširiva i prilagodljiva različitim potrebama korisnika. Međutim,

identificirana su i određena ograničenja, kao što je potreba za dodatnom optimizacijom performansi kod većeg broja korisnika i transakcija.

Rezultati pokazuju da je moguće razviti skalabilnu web trgovinu koji podržava stvaranje uplatnica, čak i u regijama gdje su tradicionalni načini plaćanja još uvijek dominantni. Ova rješenja mogu se dalje prilagođavati i poboljšavati kako bi se odgovorilo na specifične potrebe korisnika i tržišta

Razvijena web trgovina omogućuje korisnicima jednostavnu registraciju, pregled proizvoda, dodavanje proizvoda u košaricu i generiranje uplatnica za plaćanje. Sustav je dizajniran tako da bude skalabilan i siguran, koristeći moderne web tehnologije. Pri svakoj kupnji registrirani korisnici ostvaruju bodove lojalnosti koji mogu biti iskorišteni za dodatne popuste. Poseban naglasak stavljen je na sigurnost korisničkih podataka i intuitivnost korisničkog sučelja, što su ključni faktori za uspjeh u e-trgovini

Za daljnji rad na ovom području, preporuča se istraživanje novih tehnologija i pristupa za još bolju optimizaciju i sigurnost web trgovina. Također, preporuča se razvoj dodatnih modula koji bi omogućili još veće funkcionalnosti, poput napredne analitike korisničkog ponašanja i automatiziranog marketinga.

6. LITERATURA

- [1] „Online Shopping Statistics: Ecommerce Trends for [wcyear]“ [online], 05-kol-2024. Dostupno na: <https://www.tidio.com/blog/online-shopping-statistics/>. [Pristupljeno: 7.9.2024.].
- [2] „why-is-online-shopping-becoming-so-much-more-popular“ [online], 16-velj-2023. Dostupno na: <https://www.dhl.com/discover/en-nz/e-commerce-advice/e-commerce-trends/why-is-online-shopping-becoming-so-much-more-popular>. [Pristupljeno: 7.9.2024.].
- [3] W., Systems, „For building a dynamic and interactive website React.JS is the best platform“ [online], 26-lis-2020. .
- [4] „Ecommerce Success stories“ [online]. .
- [5] „<https://firebase.google.com/docs/auth>“ . .
- [6] „useState – React“ [online]. Dostupno na: <https://react.dev/reference/react/useState>. [Pristupljeno: 4.9.2024.].
- [7] M. U., Umar, „How to Generate QR Code in ReactJS“ [online], 28-velj-2024. .
- [8] „html2canvas“ [online], 22-sij-2022. Dostupno na: <https://www.npmjs.com/package/html2canvas>. [Pristupljeno: 4.9.2024.].
- [9] „How does EmailJS work?“ [online]. Dostupno na: <https://www.emailjs.com/docs/introduction/how-does-emailjs-work/>. [Pristupljeno: 7.9.2024.].

7. SAŽETAK

Razvijena je web aplikacija za e-trgovinu koristeći *React* kao *frontend* tehnologiju, *Firebase* kao *backend* uslugu i *email.js* za slanje obavijesti putem e-pošte. Glavna funkcionalnost aplikacije je omogućavanje korisnicima da kreiraju narudžbe i generiraju uplatnice. Kako bi se automatizirala komunikacija s korisnicima, implementirana je integracija s *EmailJS* uslugom za slanje e-pošte. Prilikom generiranja uplatnice, korisnik automatski prima e-poštu s detaljima o narudžbi i uplatnicom. Administrator ima mogućnost da šalje automatske e-pošte za praćenje narudžbe.

Ključne riječi: *emailjs*, *firebase*, *react*, uplatnica, web-trgovina

8. ABSTRACT

Title: Web-shop with slip payment method

An e-commerce web application was developed using React as the frontend technology, Firebase as the backend service, and Email.js for sending email notifications. The primary functionality of the application allows users to create orders and generate payment slips. To automate communication with users, an integration with the EmailJS service was implemented for sending emails. Upon generating a payment slip, the user automatically receives an email with order details and the payment slip. The administrator has the ability to send automated order tracking emails.

Keywords: emailjs, firebase, payment slip, react, webshop

9. ŽIVOTOPIS

Mislav Ostović rođen je 5. 3. 2002. u Slavonskom brodu. 2002. godine odselio se sa obitelji u Zadar gdje je 2008. godine upisao prvi razred u Osnovnoj školi Bartola Kašića. 2009. doselio se sa obitelji nazad u Slavonski brod gdje je nastavio svoje školovanje. 2016. upisao je i završio opću gimnaziju Matije Mesića. 2020. godine upisuje stručni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Trenutno studira treću godinu stručnog studija računarstva na istom fakultetu.

Potpis autora