

Web aplikacija za optičke proizvode: oglašavanje i trgovina

Vinski, Ivan

Master's thesis / Diplomski rad

2024

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:195:545523>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

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



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Sveučilišni diplomski studij informatike

Ivan Vinski

Izrada web aplikacije za oglašavanje i trgovinu optikom

Diplomski rad

Mentor: doc. dr. sc. Lucia Načinović Prskalo

Rijeka, veljača 2024.

Rijeka, 28.4.2023.

Zadatak za diplomski rad

Pristupnik: Ivan Vinski


Naziv diplomskog rada: Web aplikacija za optičke proizvode: oglašavanje i trgovina

Naziv diplomskog rada na eng. jeziku: Web application for optical products: advertising and commerce

Sadržaj zadatka: Zadatak diplomskog rada je izraditi aplikaciju za oglašavanje i trgovinu optičkih proizvoda. U radu će detaljno biti opisane sve tehnologije koje će se koristiti za izradu aplikacije te sve funkcionalnosti koje aplikacija pruža.

Mentorica:

Doc. dr. sc. Lucia Načinović Prskalo



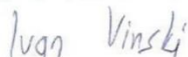
Voditeljica za diplomske radove:

Prof. dr. sc. Ana Meštrović



Zadatak preuzet: 28.4.2023.

(potpis pristupnika)



Sadržaj

Sažetak	5
1. Uvod	6
2. Metodologija	7
2.1. Agilni princip razvoja	7
2.2. Arhitektura temeljena na komponentama	7
3. Odabir tehnologija i alata	8
3.1. Visual Studio Code	8
3.2. Nuxt	10
3.3. TypeScript	10
3.4. ESLint	11
3.5. Quasar	11
3.6. Tailwind CSS	12
3.7. Firebase	13
3.7.1. Firebase Authentication	13
3.7.2. Firestore Database	14
3.7.3. Firebase Storage	14
3.7.4. Firebase Hosting	15
3.8. GitHub	15
3.9. Trustmary	16
3.10. Swiper	17
4. Razvoj web aplikacije	17
4.1. Frontend	18
4.1.1. Kreiranje projekta	18
4.1.2. Instalacija i konfiguracija paketa	19
4.2. Backend	27
4.2.1. Kreiranje projekta	27
4.2.2. Povezivanje Firebase projekta s web aplikacijom	28
4.2.3. Postavljanje Firebase autentifikacije	29
4.2.4. Postavljanje Firestore baze podataka	29
4.2.5. Postavljanje Firebase skladišta podataka	29
4.2.6. Postavljanje Firebase poslužitelja	30

5.	Pregled funkcionalnosti web aplikacije	30
5.1.	Gosti (neregistrirani korisnici).....	31
5.1.1.	Početna stranica.....	31
5.1.2.	Rezervacija besplatnog pregleda	37
5.1.3.	Stranica ponude proizvoda	39
5.1.4.	Stranica novosti.....	40
5.1.5.	Registracija i prijava	41
5.2.	Klijenti (registrirani korisnici)	43
5.2.1.	Stranica „Moj račun“	43
5.2.2.	Stranica „Košarica“.....	45
5.3.	Administratori.....	46
5.3.1.	Stranica „Općenito“	46
5.3.2.	Stranica „Poslovnice“	46
5.3.3.	Stranica „Proizvodi“	48
6.	Zaključak.....	49
7.	Popis literature i izvora.....	50
8.	Popis priloga.....	51
9.	Popis slika.....	51
10.	Popis tablica	52

Sažetak

Ovaj diplomski rad fokusira se na izradu web aplikacije za fiktivnu tvrtku „Optika Future“, specijaliziranu za prodaju optičkih proizvoda. Cilj rada je razviti sofisticiranu web platformu koja bi omogućila tvrtki da poboljša svoje poslovanje i korisničko iskustvo kroz digitalne kanale. U tu svrhu, rad detaljno istražuje različite aspekte dizajna, razvoja i implementacije web aplikacija, uključujući odabir odgovarajućih tehnologija i alata, proces razvoja web aplikacije kao i pristupe dizajniranju korisničkog sučelja. Posebna pažnja posvećena je prilagodljivosti aplikacije na različitim uređajima, sigurnosnim aspektima i osiguravanju intuitivnog korisničkog iskustva.

Glavni dijelovi rada uključuju uspješnu integraciju ključnih funkcionalnosti poput kataloga proizvoda, upravljanja košaricom, rezervacijom termina za pregled vida, pisanje i uređivanje bloga, ažuriranje kataloga proizvoda kao i implementaciju responzivnog dizajna koji osigurava optimalno korisničko iskustvo na širokom spektru uređaja. Osim toga, rad ističe važnost kontinuirane evaluacije i optimizacije web aplikacije kako bi se osigurala njena dugoročna održivost i usklađenost s promjenjivim potrebama korisnika.

Zaključno, diplomski rad demonstrira kako temeljit pristup razvoju web aplikacije može značajno doprinijeti digitalnoj transformaciji i konkurentnosti poslovanja u optičkoj industriji.

Ključne riječi: web aplikacija, frontend, backend, nuxt, tailwind css, firebase, responzivnost

1. Uvod

Današnje poslovanje sve više prelazi u digitalni svijet, gdje online oglašavanje i trgovina postaju ključni čimbenici uspjeha za mnoge tvrtke. Razvijanje poslovne web aplikacije za integrirano upravljanje oglašavanjem i trgovinom postaje imperativ kako bi tvrtke mogle učinkovito dosegnuti svoje ciljne skupine i optimizirati svoje poslovanje.

Tema ovog diplomskog rada je izrada web aplikacije za fiktivnu firmu „Optika Future“, koja se bavi prodajom optičkih proizvoda poput naočala, leća i drugih optičkih dodataka. Zadatak ovog rada je razvoj kompleksne web aplikacije koja će olakšati poslovanje Optike Future, pružiti korisnicima jednostavan način za pregled i kupovinu proizvoda te poboljšati sveukupno iskustvo kupaca.

Motivacija iza odabira ove teme proizlazi iz rastuće potrebe za online prisustvom trgovina, uključujući i trgovine optičkim proizvodima. Kako bi firma ostala konkurentna i zadovoljila potrebe suvremenih potrošača, ključno je imati dobro dizajniranu web aplikaciju koja će olakšati proces kupovine, pružiti informacije o proizvodima i uslugama te omogućiti učinkovitu komunikaciju s kupcima.

Ciljevi ovog diplomskog rada obuhvaćaju opis različitih aspekata izrade web aplikacije. U prvom redu, fokus je stavljen na razvoj funkcionalne web aplikacije koja će podržavati online trgovinu. To uključuje implementaciju svih potrebnih funkcionalnosti, poput pregleda i ažuriranja kataloga proizvoda, upravljanja košaricom, te efikasnog procesa naručivanja i plaćanja.

Drugi ključni cilj diplomskog rada je dizajniranje intuitivnog korisničkog sučelja. Ova komponenta web aplikacije treba omogućiti korisnicima jednostavno snalaženje, pregled proizvoda i brzu kupovinu.

Važan aspekt je i prilagodba web aplikacije različitim uređajima, uključujući računala, tablete i mobilne telefone. Responzivnost web sučelja ključno je za pružanje ugodnog iskustva svim korisnicima, neovisno o platformi s koje korisnik pristupa web aplikaciji.

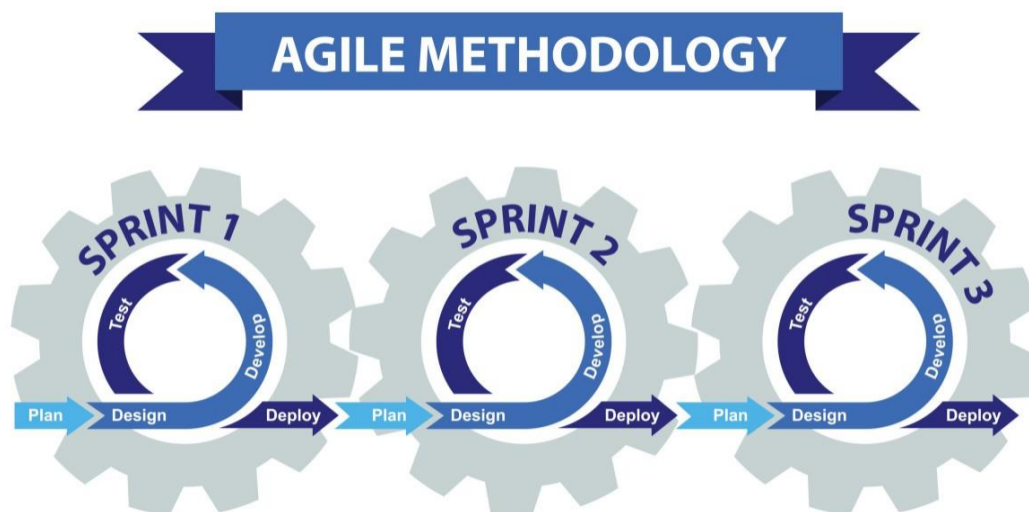
Ovaj rad obuhvaća pregled metodologija, detaljnu analizu i odabir tehnologija i alata koji su ključni za razvoj web aplikacije, uključujući, ali ne ograničavajući se na Visual Studio Code, Nuxt, Quasar Framework, Tailwind CSS, Firebase i druge. Kroz razvoj frontend i backend dijelova aplikacije, rad će prikazati kako se integriraju različite komponente i tehnologije kako bi se stvorila funkcionalna, sigurna i korisniku prijateljska web aplikacija. Sam kraj rada obuhvatit će pregled osnovnih funkcionalnosti gotove web aplikacije, uz poseban naglasak na različite korisničke uloge.

2. Metodologija

Za razvoj web aplikacije koristit će se agilni razvojni procesi, s naglaskom na iterativnom razvoju, fleksibilnosti i stalnom testiranju. Uz agilni razvoj koristit će se i arhitektura temeljena na komponentama koja dodatno pridonosi fleksibilnost u razvoj web aplikacije.

2.1. Agilni princip razvoja

Agilni razvojni proces naglašava fleksibilnost i brzu adaptaciju na promjene kroz iterativni razvoj, gdje se projekti razvijaju u kratkim ciklusima zvanim sprintovi. Ovaj pristup omogućuje timovima da kontinuirano poboljšavaju proizvod, brzo reagiraju na promjene i učinkovito surađuju. Iterativni razvoj razbija projekte na manje dijelove koji se razvijaju u više faza, omogućavajući brzu isporuku funkcionalnosti i prilagodbu na temelju povratnih informacija. Fleksibilnost je ključna, omogućujući prilagodbe planova i proizvoda bez značajnog utjecaja na projekt, što je idealno za dinamično tehnološko okruženje. Slika 1 ilustrira proces agilnog razvoja.



Slika 1 Ilustracija agilnog razvojnog procesa (Technologies 2023)

2.2. Arhitektura temeljena na komponentama

Arhitektura temeljena na komponentama (engl. Component-Based Architecture) predstavlja suvremen pristup razvoju softvera, koji se temelji na ideji razgradnje kompleksnih aplikacija na manje, samostalne jedinice poznate kao komponente. Ovaj pristup omogućuje developerima da izgrade aplikacije s većom modularnošću, lakšim održavanjem i boljom mogućnošću ponovne uporabe koda. U kontekstu web razvoja, komponente se često definiraju kao skupovi HTML-a, CSS-a i JavaScripta koji zajedno tvore zasebnu funkcionalnu jedinicu, poput gumba, navigacijske trake ili obrasca za unos. (Nandaniya 2023).

Prednosti ove arhitekture su:

- Modularnost – Aplikacije se mogu razvijati, testirati, održavati i ažurirati na razini pojedinačnih komponentata, što pojednostavljuje proces razvoja i smanjuje potencijal za greške.
- Ponovna upotrebljivost – Komponente su dizajnirane tako da budu ponovno upotrebljive u različitim dijelovima aplikacije ili čak u različitim projektima, što vodi do uštede vremena i resursa.
- Skalabilnost – Aplikacije zasnovane na komponentama lakše se skaliraju jer se nove funkcionalnosti mogu dodavati kao nove komponente bez ometanja postojećeg koda.
- Lako održavanje – Ažuriranja i ispravci mogu se provoditi na razini komponente bez potrebe za prepravljajem cijele aplikacije, što olakšava održavanje i smanjuje mogućnost pojave grašaka kod promjena
- Zajednica i resursi – Popularni okviri poput Vue.js i React.js podržavaju arhitekturu zasnovanu na komponentama i imaju velike zajednice, što znači da su resursi, dokumentacija i podrška lako dostupni.

U kontekstu razvoja ove web aplikacije, korištenje arhitekture temeljene na komponentama omogućuje efikasno upravljanje složenošću projekta, brzo reagiranje na zahtjeve za promjenama i osiguranje visoke kvalitete krajnjeg proizvoda. Nuxt, kao razvojni okvir baziran na Vue.js-u, pruža idealnu platformu za implementaciju ove arhitekture zahvaljujući svojoj podršci za komponentno orijentiran razvoj web aplikacija.

3. Odabir tehnologija i alata

U današnjem dinamičnom svijetu informatike, odabir pravilnih tehnologija i alata bitan je korak u procesu razvoja web aplikacije. S obzirom na raznovrsnost jezika, okvira i alata dostupnih na tržištu, programeri se suočavaju s izazovom pronalaženja optimalnog skupa tehnologija koji će zadovoljiti specifične zahtjeve projekta. Ovaj proces odabira nije samo tehnički, već uključuje i razmatranje faktora kao što su performanse, skalabilnost, sigurnost, zajednica podrške te troškovi održavanja. U ovom poglavlju ćemo istražiti ključne aspekte koje treba uzeti u obzir prilikom odabira tehnologija i alata za razvoj web aplikacije, istražujući kako pravilna selekcija može značajno utjecati na uspjeh projekta te olakšati proces razvoja i održavanja aplikacije u budućnosti.

3.1. Visual Studio Code

Odabir odgovarajućeg uređivača koda prvi je korak u postizanju efikasnog i produktivnog programiranja. Na tržištu postoji mnogo opcija, uključujući Sublime Text, Atom, Brackets, IntelliJ IDEA, Eclipse, i mnoge druge, svaka s vlastitim skupom funkcionalnosti, korisničkim sučeljem i ekosustavom dodataka. Međutim, Visual Studio Code (poznatiji kao VSCode), besplatan uređivač koda razvijen od strane Microsofta, ističe se svojom sveobuhvatnošću, prilagodljivošću i snažnom podrškom za razvoj.

VSCode je postao izuzetno popularan među developerima zbog svoje lakoće korištenja, visokih performansi, i bogatog skupa značajki, uključujući podršku za gotovo sve programske jezike, ugrađeni terminal, duboku integraciju s Gitom, te širok spektar dostupnih proširenja koja omogućavaju prilagodbu

uređivača specifičnim potrebama projekta. Njegova sposobnost prilagodbe omogućava programerima da konfiguriraju radno okruženje u skladu sa svojim preferencijama, što može povećati učinkovitost i smanjiti vremenski okvir razvoja. Osim toga, zajednica koja stoji iza VSCode-a neprestano radi na poboljšanju i dodavanju novih funkcionalnosti, osiguravajući da alat ostane na vrhu tehnoloških inovacija i najboljih praksi u industriji razvoja softvera.

Jedna od glavnih prednosti korištenja VSCode-a je njegova IntelliSense funkcionalnost. Prema VSCode korisničkom priručniku IntelliSense je opći pojam za različite značajke uređivanja koda koje uključuju automatsko dovršavanje koda, informacije o parametrima funkcija, brze informacije te liste članova (VSCode, 2024). Ova funkcionalnost može značajno ubrzati proces razvoja, omogućavajući programerima da pišu kod brže i s manje grešaka. Ugrađeni program za pronalaženje pogrešaka (engl. debugger) u VSCode-u još je jedan moćan alat koji programerima omogućava da učinkovito identificiraju i rješavaju greške unutar koda, što je ključno za održavanje visoke kvalitete aplikacije.

S obzirom na navedene značajke, VSCode se pokazao kao odličan izbor za potrebe ovog projekta. Integracijom naprednih alata i proširenja kao što su Prettier, Nuxt, Tailwind CSS IntelliSense i Volar osigurana je dodatna učinkovitost i dosljednost u razvoju ovog projekta.

Prettier je popularan alat za formatiranje koda koji podržava mnoge jezike i integrira se s većinom uređivača koda, uključujući VSCode. Njegova primarna funkcija je automatsko formatiranje koda prilikom spremanja datoteka prema unaprijed definiranim pravilima stila, što osigurava dosljednost koda i olakšava čitanje i razumijevanje koda među različitim članovima tima. Prettier pomaže u održavanju čistog koda tako što uklanja originalni stil pisanja i primjenjuje svoj set pravila, što rezultira jedinstvenim i konzistentnim stilom koda neovisno o autoru. (Long, 2017).

Nuxt je dodatak za VSCode koji je posebno dizajniran da poboljša iskustvo rada s Nuxt projektima unutar ovog popularnog uređivača koda. Ovaj dodatak nudi niz funkcionalnosti koje olakšavaju razvoj, uključujući stvaranje projekata, upravljanje direktorijima i datotekama, upravljanje ovisnostima (engl. dependencies), IntelliSense i automatsko dovršavanje koda, prilagođene isječke (engl. snippets) za Nuxt, Nitro i Vue datoteke, kao i personalizirane predloške za Vue datoteke. (Farrag 2024)

Tailwind CSS IntelliSense je dodatak za VSCode koji znatno poboljšava iskustvo razvoja s Tailwind CSS-om. Pruža napredne značajke poput automatskog dovršavanja koda, isticanje sintakse i provjere ispravnosti koda (engl. linting). Automatsko dovršavanje omogućuje inteligentne prijedloge za nazive klasa te CSS funkcije i direktive, dok provjera ispravnosti ističe moguće greške i probleme u CSS-u i oznakama. Dodatak također omogućava pregledavanje kompletnog CSS-a za Tailwind klasu postavljanjem pokazivača iznad nje te nudi alternativni način rada s CSS-om u VSCode-u koji osigurava podršku za IntelliSense čak i kada se koriste specifična Tailwind pravila. (Tailwind Labs 2024)

Volar je službeni dodatak za VSCode namijenjen poboljšanju razvojnog iskustva s Vue.js-om. Ovaj dodatak pruža napredne mogućnosti kao što su isticanje sintakse i analiza koda te je posebno dizajniran da podrži jednostranične komponente (engl. Single File Components, SFCs) Vue.js-a. Volar se ističe svojom izvedbom i arhitekturom, zbog čega je postao novi službeni dodatak za Vue.js, zamijenivši prethodni Vetur dodatak za projekte koji koriste Vue 2. Volar pruža podršku za TypeScript unutar Vue SFC-ova, zajedno s mnogim drugim značajkama, što ga čini ključnim alatom za razvoj u Vue.js ekosustavu. (Chu 2023)

3.2. Nuxt

„Nuxt je okvir otvorenog koda (engl. open-source) dizajniran za pojednostavnjenje stvaranja „full-stack“ web aplikacija i web stranica koristeći Vue.js. Ovaj okvir nudi fleksibilnost, automatizaciju i različite načine prikaza za optimalnu izvedbu“ („Nuxt Overview“ 2024). Dizajniran je za stvaranje modernih web aplikacija, dok istovremeno olakšava razvoj univerzalnih ili jednostraničnih (engl. single-page) Vue.js aplikacija kroz svoj moćan i fleksibilan set značajki.

Jedna od glavnih prednosti Nuxt-a je njegova sposobnost iscrtavanja na strani poslužitelja (engl. Server-Side Rendering, SSR), koja omogućava aplikacijama da se učitavaju brže i poboljšavaju optimizaciju za tražilice (engl. Search Engine Optimization, SEO). SSR osigurava da su sadržaj i struktura stranice dostupni pretraživačima u trenutku indeksiranja, što je ključno za poboljšanje vidljivosti web stranica u rezultatima pretraživanja.

Nuxt također pruža automatizirano dijeljenje koda (engl. code-splitting), što znači da se skripte dijele na manje pakete koji se učitavaju samo kada su potrebni. Ovo poboljšava performanse aplikacije smanjenjem vremena učitavanja.

Za ovaj projekt, Nuxt je odabran zbog svoje modularnosti i lakoće integracije s ostalim tehnologijama i alatima. Nuxt nudi bogat ekosustav modula i dodataka koji omogućavaju jednostavnu integraciju funkcionalnosti poput autentifikacije, podrške za progresivne web aplikacije (engl. Progressive Web Apps, PWA), i automatskog generiranja ruta, što značajno ubrzava proces razvoja.

Osim toga, Nuxt pruža izvrsnu podršku za Vue.js, omogućavajući programerima da iskoriste sve prednosti Vue-a, uključujući reaktivnost, komponentni sustav i direktive, uz dodatne pogodnosti koje nudi Nuxt. Ovo uključuje predloške stranica, automatsko generiranje meta tagova za SEO, i lako upravljanje stanjem aplikacije pomoću Nuxt *useState()* funkcionalistima.

Važna karakteristika Nuxt-a koja je bila presudna za njegov odabir u ovom projektu je njegova prilagodljivost različitim tipovima projekata, od jednostraničnih aplikacija (engl. Single Page Applications, SPA), preko SSR aplikacija, do statički generiranih stranica. Ova fleksibilnost omogućava Nuxt-u da odgovori na različite potrebe projekta, pružajući temelj za izgradnju brze, sigurne i korisnički privlačne web aplikacije.

Uzimajući u obzir ove prednosti, Nuxt se nametnuo kao prirodan izbor za ovaj projekt, nudeći kombinaciju performansi, SEO optimizacije, i razvojne efikasnosti, što je ključno za postizanje ciljeva projekta.

3.3. TypeScript

„TypeScript je nadskup JavaScripta koji dodaje opcionalno statičko tipiziranje i napredne značajke JavaScriptu. Razvijen je od strane Microsofta, a počeo se koristiti u listopadu 2012. godine“ (Olawanle 2023). To je zapravo snažan programski jezik koji omogućava programerima da pišu sigurniji i razumljiviji kod. Uvođenjem statičkog tipiziranja, TypeScript pruža alate za definiranje strukture podataka i očekivanih tipova varijabli, funkcija i objekata, što doprinosi ranijem otkrivanju pogrešaka, boljoj integraciji s razvojnim okruženjima i efikasnoj suradnji unutar timova. Osim toga, TypeScript donosi napredne značajke

kao što su sučelja (engl. interface), generički tipovi (engl. generic types), nabiranja (engl. enumeration), i više, čime se obogaćuju mogućnosti JavaScript-a i omogućuje razvoj složenijih aplikacija.

Jedna od ključnih prednosti TypeScript-a je njegova kompatibilnost s postojećim JavaScript kodom, što znači da se bilo koji validan JavaScript kod može smatrati validnim TypeScript kodom. Ovo omogućava laganu integraciju TypeScript-a u postojeće projekte bez potrebe za prepravkom cijelog koda. Nakon kompilacije, TypeScript se prevodi u čisti JavaScript, čineći kod izvršivim na bilo kojoj platformi koja podržava JavaScript, uključujući web preglednike i Node.js okruženja.

S obzirom na sve veću složenost frontend i backend JavaScript aplikacija, TypeScript se sve češće koristi u razvoju, nudeći robustan alat za upravljanje tim složenostima i poboljšanje kvalitete koda. Njegova popularnost kontinuirano raste među razvojnim zajednicama, a koristi se u razvoju mnogih poznatih okvira i biblioteka kao što su Angular, NestJS i mnogi drugi, potvrđujući svoje mjesto kao ključna tehnologija u modernom web razvoju.

3.4. ESLint

„ESLint je alat otvorenog koda za provjeru kvalitete JavaScript koda koji je izvorno stvorio Nicholas C. Zakas u lipnju 2013. godine. Često se koristi za pronalaženje problematičnih uzoraka ili koda koji se ne pridržava određenih stilskih smjernica“ (Gupta 2021). Ovaj alat omogućava programerima da definiraju i primjenjuju pravila (engl. rules) za stil i praksu kodiranja u JavaScriptu. ESLint analizira kod u stvarnom vremenu tijekom razvoja, identificira potencijalne probleme i upozorava programere na potencijalne greške ili odstupanja od definiranih pravila.

Jedna od ključnih prednosti ESLint-a je njegova prilagodljivost. Programeri mogu konfigurirati ESLint prema specifičnim potrebama projekta, definirajući vlastita pravila ili koristeći gotove predloške (engl. presets) za popularne stilove kodiranja poput Airbnb, Standard, ili drugih. Ovaj pristup osigurava da se kodiranje provodi u skladu s odabranim konvencijama i standardima.

ESLint također pruža mogućnost automatskog ispravljanja pronađenih problema, olakšavajući programerima u uvođenju kvalitete u kod bez ručnog ispravljanja. Integriran je s mnogim razvojnim okruženjima i uređivačima koda, uključujući VSCode, što omogućava brzo otkrivanje i ispravljanje problema dok programeri pišu kod.

Korištenje ESLint-a u ovom projektu poboljšat će se kvaliteta koda osiguravajući dosljednost otkrivanjem grešaka u ranim fazama razvoja. Alat će automatski provjeriti kod s definiranim pravilima stilova i najboljih praksi, što pomaže u održavanju visokog standarda kodiranja.

3.5. Quasar

„Quasar je popularan okvir otvorenog koda za izradu web aplikacija visokih performansi i mobilnih aplikacija koristeći jedinstvenu bazu koda. Izgrađen je na temelju Vue.js, progresivnog JavaScript okvira, i pruža sveobuhvatan skup alata i komponenti za stvaranje modernih i responzivnih aplikacija“ (Modi 2023).

Quasar je dizajniran s fokusom na performanse i efikasnost, omogućavajući brz razvoj visokokvalitetnih aplikacija s manje koda. Jedna od ključnih prednosti Quasar okvira je njegov sveobuhvatan set UI komponenata koje su optimizirane za performanse i prilagodljive na različitim platformama. Ove komponente su dizajnirane da rade bez problema na svim uređajima, pružajući dosljedno i privlačno korisničko iskustvo bez obzira na veličinu ekrana ili operacijski sustav.

Za ovaj projekt, izbor Quasar okvira bio je motiviran željom za stvaranjem jedinstvenog i prilagodljivog korisničkog sučelja koje se može efikasno koristiti na web platformi. Iako je ovaj projekt fokusiran isključivo na razvoj web aplikacije, Quasar-ova sposobnost da iz jedinstvenog izvornog koda omogući razvoj kvalitetnih aplikacija ne samo za web, već i za mobilne i desktop platforme, predstavlja važnu prednost. Ova fleksibilnost omogućava potencijalno proširenje projekta u budućnosti, omogućavajući „Optici Future“ da lako prilagodi svoju web aplikaciju za različite platforme bez potrebe za ponovnim pisanjem koda.

Korištenjem Quasar okvira, maksimalno je iskorišten ekosustav Vue.js-a za razvoj web aplikacije, istovremeno zadržavajući mogućnost budućeg proširenja na mobilne i desktop platforme bez dodatnih troškova i složenosti. Ova strategija ne samo da olakšava trenutni razvoj, već i postavlja temelje za skalabilnost i prilagodljivost aplikacije u skladu s budućim poslovnim potrebama „Optike Future“.

Pristup „napiši jednom, primijeni posvuda“ (engl. „write once, deploy anywhere“, WODA) koji nudi Quasar smanjuje kompleksnost razvojnog procesa i ubrzava izdavanje aplikacije, čime se optimiziraju resursi i vrijeme potrebno za razvoj. Integracija s različitim frontend i backend tehnologijama olakšava implementaciju kompleksnih funkcionalnosti potrebnih za ovu web aplikaciju, čineći Quasar Framework ključnim elementom u ostvarivanju projektnih ciljeva i osiguravanju temelja za buduće tehnološko napredovanje.

3.6. Tailwind CSS

Tailwind CSS je „utility-first“ CSS okvir koji omogućava izradu prilagođenih korisničkih sučelja bez potrebe za pisanjem CSS-a od nule. Za razliku od tradicionalnih CSS okvira poput Bootstrap-a ili Foundation-a, koji nude predefinirane komponente i stilove, Tailwind pruža niz pomoćnih (engl. utility) klasa koje se mogu primjenjivati izravno u HTML elemente kako bi se brzo stiliziralo sučelje.

Pristup „utility-first“ podrazumijeva korištenje malih, ponovno upotrebljivih klasa koje implementiraju pojedinačna CSS svojstva (npr. margine, boje, fontove). Ovo omogućava finu kontrolu nad dizajnom i lako prilagođavanje.

Jedna od ključnih prednosti Tailwind CSS-a je njegova modularnost i skalabilnost. Razvojni programeri mogu jednostavno dodavati, mijenjati ili uklanjati stilove bez straha od narušavanja postojećeg dizajna, što Tailwind čini izuzetno prilagodljivim za projekte različitih veličina i složenosti. Tailwind također potiče konzistentnost u dizajnu kroz cijeli projekt, zahvaljujući sustavu konfiguracije koji omogućava definiranje globalnih stilova kao što su boje, fontovi, razmaci i slično.

Za ovaj projekt, upotreba Tailwind CSS-a odabrana je zbog potrebe za visokom razinom prilagodljivosti i efikasnosti u razvoju korisničkog sučelja. Tailwind-ov „utility-first“ pristup omogućio je brzu i iterativnu implementaciju korisničkog sučelja, osiguravajući da web aplikacija ne samo da izgleda privlačno, već i pruža izvrsno korisničko iskustvo. Sposobnost brzog razvoja i prilagodbe stilova bez potrebe za pisanjem

velikih količina CSS koda učinila je Tailwind nezamjenjivim alatom u razvoju kako ove tako i većinu današnjih modernih web aplikacija.

Važno je napomenuti kako je u ovom projektu Tailwind CSS korišten za prilagodbu dizajna, dok je Quasar Framework pružao funkcionalne komponente poput tablica, obrazaca, padajućih izbornika i drugo, osiguravajući funkcionalnost i vizualnu privlačnost sučelja. Ova kombinacija omogućila je brz razvoj jedinstvenog, prilagodljivog sučelja bez kompromisa na funkcionalnosti.

3.7. Firebase

„Firebase je sveobuhvatna platforma za razvoj mobilnih i web aplikacija koju je razvio Google. Pruža programerima niz alata i usluga temeljenih na oblaku dizajniranih da pomognu u stvaranju, održavanju i unaprjeđenju aplikacija“ (Özşahan 2023). Ova platforma omogućava programerima da brzo izgrade i skaliraju svoje aplikacije bez potrebe za upravljanjem infrastrukturom. Firebase nudi različite funkcionalnosti, uključujući ali ne ograničavajući se na autentifikaciju korisnika (Firebase Authentication), pohranu podataka u realnom vremenu (Firestore Database i Realtime Database), pohranu datoteka (Firebase Storage), hosting web aplikacija (Firebase Hosting), i još mnogo toga.

Jedna od ključnih prednosti Firebase-a je njegova integracija s Google Cloud Platformom, što omogućava pristup dodatnim uslugama kao što su računalni resursi, analitika i strojno učenje. Firebase također nudi moćne alate za analizu i poboljšanje korisničkog iskustva, poput Firebase analitike, koja pruža uvide u ponašanje korisnika i efikasnost aplikacije.

Firebase je odabran zbog svoje jednostavne upotrebe, skalabilnosti i integracije s različitim platformama i jezicima. Platforma omogućava brz razvoj s bogatim setom gotovih funkcionalnosti, smanjujući tako vrijeme potrebno za razvoj i omogućavajući fokus na ključne značajke aplikacije. Korištenjem Firebase-a, omogućena je brza implementacija autentifikacije, jednostavno upravljanje bazom podataka, besplatno posluživanje aplikacije putem Firebase Hosting-a te analiza korisničkog ponašanja, čime se osigurava visokokvalitetno korisničko iskustvo i pouzdanost aplikacije.

3.7.1. Firebase Authentication

Firebase autentifikacija (engl. authentication) je usluga unutar Firebase platforme koja pruža jednostavan i siguran način za autentifikaciju korisnika u mobilnim i web aplikacijama. Ova usluga podržava razne metode prijave, uključujući email i lozinku, društvene mreže kao što su Google, Facebook i Twitter, i autentifikaciju pomoću telefonskog broja. Firebase autentifikacija također podržava anonimnu autentifikaciju, što omogućava korisnicima da koriste aplikaciju bez stvaranja stalnog računa.

Jedna od glavnih prednosti Firebase autentifikacije je njegova integracija s ostalim Firebase uslugama, što omogućava razvojnim programerima da lako upravljaju pristupom i podacima korisnika. Na primjer, može se koristiti zajedno s Firestore bazom podataka, pisanjem sigurnosnih pravila koja kontroliraju pristup podacima na temelju statusa autentifikacije korisnika.

Upotreba Firebase autentifikacije u ovom projektu omogućila je brzo implementiranje sigurnog i pouzdanog sustava prijave, smanjujući potrebu za razvojem kompleksnih autentifikacijskih sustava od nule. Ovo ne samo da ubrzava proces razvoja, već i poboljšava korisničko iskustvo nudeći korisnicima različite opcije za prijavu, čime se povećava angažman i zadržavanje korisnika. Osim toga, Firebase autentifikacija pruža napredne značajke poput autentifikacije u dva koraka (engl. two-factor authentication) i resetiranja lozinke, dodatno povećavajući sigurnost aplikacije.

3.7.2. Firestore Database

Firestore baza podataka (engl. database), poznata i kao Cloud Firestore, je fleksibilna, skalabilna baza podataka za mobilne i web aplikacije iz Firebase i Google Cloud Platforme. Firestore omogućuje pohranu i sinkronizaciju podataka između korisnika u realnom vremenu, čime olakšava izgradnju interaktivnih aplikacija. Ova ne relacijska (NoSQL) baza podataka strukturirana je oko kolekcija i dokumenata, gdje svaka kolekcija može sadržavati više dokumenata, a svaki dokument može sadržavati različite tipove podataka kao što su brojevi, tekst, mape, nizovi i slično.

Jedna od ključnih karakteristika Firestore-a je njegova sposobnost da automatski upravlja složenim upitima i indeksiranjem, što omogućava programerima da lako pristupaju i manipuliraju podacima. Firestore također nudi robusne sigurnosne značajke koje omogućavaju definiranje detaljnih pravila pristupa na temelju identiteta korisnika, osiguravajući da podaci ostanu sigurni i privatni.

Firestore je odabran zbog svoje sposobnosti da osigura visoku razinu skalabilnosti i pouzdanosti, kao i zbog jednostavnosti integracije s ostalim Firebase uslugama. Korištenjem Firestore-a, omogućena je jednostavna sinkronizacija podataka korisnika u realnom vremenu, što je ključno za pružanje besprijekornog korisničkog iskustva, bilo da se radi o ažuriranju inventara proizvoda, praćenju narudžbi ili omogućavanju komunikacije uživo između korisnika. Ova baza podataka omogućava fokus na razvoj ključnih funkcionalnosti aplikacije, umjesto da se troši vrijeme na upravljanje infrastrukturom baze podataka.

3.7.3. Firebase Storage

Firestore pohrana (engl. storage) je moćna i sigurna usluga za pohranu datoteka namijenjena Firebase aplikacijama. Omogućava pohranu i posluživanje korisničkih sadržaja kao što su slike, videozapisi, audio zapisi i bilo koja druga vrsta datoteka. Firebase Storage posebno je dizajniran za skalabilnost i jednostavnu integraciju s mobilnim i web aplikacijama, pružajući pouzdanu infrastrukturu za upravljanje velikim količinama podataka i intenzivni promet datoteka.

Jedna od ključnih prednosti Firebase pohrane je njegova integracija s Firebase autentifikacijom, što omogućava programerima da lako postavite sigurnosna pravila za pristup i upravljanje datotekama na temelju identiteta korisnika. Ovo osigurava da su datoteke zaštićene i dostupne samo ovlaštenim korisnicima, čime se povećava sigurnost i privatnost podataka.

Firestore pohrana koristi Google Cloud Storage u pozadini, što osigurava visoku dostupnost i globalnu mrežu za brzo posluživanje sadržaja, neovisno o lokaciji korisnika. Osim toga, Firestore pohrana nudi

robusne mogućnosti za upravljanje datotekama, uključujući otpremu i preuzimanje datoteka u pozadini, pauziranje i nastavak prijenosa, te praćenje napretka prijenosa, što poboljšava korisničko iskustvo.

Firestore pohrana odabrana je zbog svoje efikasnosti u upravljanju medijskim datotekama i ostalim sadržajima. Korištenjem Firestore pohrane, osigurano je sigurno pohranjivanje i brzo posluživanje fotografija proizvoda, promotivnih materijala i drugih relevantnih datoteka korisnicima aplikacije. Ova usluga omogućava aplikaciji da lako upravlja velikim količinama korisničkog sadržaja, pružajući glatko i učinkovito korisničko iskustvo bez kompromisa na performansama ili skalabilnosti.

3.7.4. Firebase Hosting

Firebase Hosting je usluga koja pruža brz i siguran hosting za web aplikacije. Omogućava programerima da jednostavno postavite statičke sadržaje kao što su HTML, CSS, JavaScript i druge datoteke, te ih poslužuju putem globalne mreže za dostavljanje podataka (engl. Content Delivery Network, CDN). „CDN je distribuirani sustav poslužitelja dizajniran za što bržu isporuku web sadržaja krajnjim korisnicima“ (Fitzgerald 2024). Jedna od ključnih karakteristika Firebase Hostinga je njegova sposobnost da osigura besprijekornu isporuku sadržaja s niskom latencijom, neovisno o geografskoj lokaciji korisnika.

Firebase Hosting integriran je s ostalim Firebase uslugama, što omogućava laku implementaciju dinamičkih funkcionalnosti poput autentifikacije korisnika, rada s bazom podataka i analitike unutar statičkih web aplikacija. Osim toga, Firebase Hosting podržava automatsko SSL šifriranje, osiguravajući da su sve veze sigurne i štite podatke korisnika.

Jednostavnost korištenja jedna je od glavnih prednosti Firebase Hostinga. Programeri mogu brzo postaviti i ažurirati svoje aplikacije koristeći Firebase CLI alate, omogućavajući brze iteracije i kontinuirano isporučivanje novih verzija aplikacije. Firebase Hosting također nudi funkcionalnosti kao što su praćenje verzije izdanja, pregledi uživo prije službenog lansiranja i prilagođene domene, što dodatno poboljšava upravljanje i distribuciju web aplikacija.

U kontekstu razvoja ovog projekta, korištenje Firebase Hostinga omogućilo je efikasno postavljanje i skaliranje web aplikacije, pružajući brz pristup sadržaja svim korisnicima. Ova usluga je odabrana zbog svoje integracije s ostalim Firebase alatima, sigurnosti i pouzdanosti, čime se osigurava brzo i sigurno distribuiranje aplikacije korisnicima, uz održavanje visokih standarda performansi i sigurnosti.

3.8. GitHub

GitHub je esencijalan alat u svijetu razvoja softvera, posebno u kontekstu razvoja web aplikacija. Kao vodeća platforma za kontrolu verzija i kolaboraciju, GitHub omogućava programerima da učinkovito upravljaju izvornim kodom svojih projekata koristeći Git, distribuirani sustav kontrole verzija. Osnovan 2008. godine, GitHub je brzo postao nezamjenjiv resurs za milijune developera širom svijeta, pružajući snažne funkcionalnosti poput grananja (engl. branching), zahtjeva za povlačenjem (engl. pull requests) i pregleda koda (engl. code reviews), koji olakšavaju kolaborativni rad i doprinose visokoj kvaliteti softverskih projekata.

Korištenje GitHub-a u razvoju web aplikacija posebno dolazi do izražaja kroz njegovu sposobnost olakšavanja kontinuirane integracije i kontinuirane isporuke (engl. Continuous Integration and Continuous Delivery, CI/CD), omogućavajući timovima da automatiziraju testiranje i distribuciju aplikacija. „Kontinuirana integracija je filozofija kodiranja i skup praksi koji potiču razvojne timove da često implementiraju male promjene u kodu i provjeravaju ih u repozitoriju za kontrolu verzija. Kontinuirana isporuka nastavlja tamo gdje kontinuirana integracija završava i automatizira isporuku aplikacija u odabrana okruženja, uključujući produkcijsko, razvojno i testno okruženje. Kontinuirana isporuka je automatizirani način za dodavanje promjena koda u tim okruženjima“ (Sacolick 2023).

GitHub također potiče otvorenu suradnju putem svojih bogatih zajednica, gdje programeri mogu dijeliti kod, koristiti kod druge zajednice kroz grananje (engl. forking) i doprinosti projektima otvorenog koda putem „pull request“ opcije, čime se potiče inovacija i brži razvoj. S obzirom na te karakteristike, GitHub ne samo da služi kao platforma za upravljanje kodom, već i kao središnje mjesto za razmjenu znanja i iskustava unutar globalne developerske zajednice.

Važno je napomenuti da, iako GitHub nudi besplatne račune za pojedince i javne projekte, postoje i plaćeni planovi za privatne repozitorije i dodatne funkcionalnosti koje koriste poduzeća. GitHub također integrira razne dodatne usluge i alate koji dodatno proširuju njegove mogućnosti, uključujući upravljanje projektima, praćenje problema (engl. issue tracking) i wiki stranice, što ga čini izuzetno fleksibilnim alatom prilagodljivim različitim potrebama projekata razvoja web aplikacija.

U ovom projektu, GitHub je poslužio kao centralni alat za upravljanje izvornim kodom i organizaciju razvojnog procesa. Omogućio je strukturiranje koda, učinkovito praćenje izmjena i dokumentiranje svih promjena. Upotreba GitHub-a doprinijela je visokoj kvaliteti koda i omogućila neprekidno unapređivanje aplikacije, istovremeno osiguravajući transparentnost rada i olakšavajući pregled povijesti razvoja aplikacije.

3.9. Trustmary

Trustmary je sveobuhvatna platforma za upravljanje recenzijama koja omogućava poduzećima da poboljšaju svoje online prisustvo i povećaju povjerenje među korisnicima. Ključne značajke uključuju mogućnost prikupljanja i prikazivanja autentičnih recenzija korisnika na web stranicama. Trustmary nudi jednostavne metode za uvoz postojećih recenzija s različitih platformi poput Google-a, Facebook-a, Yelp-a i drugih, kao i automatizaciju procesa prikupljanja novih recenzija. Osim toga, platforma nudi napredne opcije za prilagodbu i integraciju sa različitim sustavima upravljanja sadržajem (engl. Content Management System, CMS), čineći je kompatibilnom s popularnim platformama kao što su WordPress, Shopify, Squarespace i Wix.

Trustmary pomaže poduzećima da izgrade digitalno povjerenje ističući pozitivne recenzije koje potiču odluke o kupnji, poboljšavaju SEO i povećavaju konverzije na web stranicama. Pod konverzije smatramo konverziju posjetitelja (osobe koja samo pregledava sadržaj) u kupca. Platforma nudi i mogućnosti praćenja uspjeha putem ugrađenih alata za praćenje ciljeva, omogućujući korisnicima da vide stvarne rezultate koje recenzije donose njihovoj web stranici.

3.10. Swiper

„Swiper je najmoderniji, besplatni softver otvorenog koda prilagođen za rad s uređajima osjetljivim na dodir s tranzicijama ubrzanim hardverom i izvanrednim izvornim ponašanjem“ (Swiper, 2024).

Jedna od istaknutih karakteristika je responzivnost, omogućujući jednostavno prilagođavanje klizača različitim veličinama ekrana. Bez obzira na to koristi li se na desktop, tablet ili mobilnim uređajima, Swiper osigurava dosljedno i fluidno iskustvo korisnika.

Osim responzivnosti, Swiper se ističe svojom podrškom za rad s uređajima osjetljivim na dodir, što ga čini idealnim rješenjem za mobilne aplikacije. Korisnici mogu jednostavno klizati prstom po zaslonu kako bi pregledavali sadržaj klizača, što pridonosi intuitivnom interakcijskom iskustvu.

Swiper pruža fleksibilnost kroz podršku različitim vrstama klizača. Bez obzira želite li klizače s jednim ili više slajdova odjednom, s petljom, automatskom reprodukcijom ili nešto drugo, Swiper to omogućuje. Dodatak za virtualne slajdove olakšava rad s velikim skupom podataka, čime se smanjuje potreba za učitavanjem svih slajdova odjednom.

Osim toga, Swiper nudi obilje opcija i prilagodbi, uključujući podešavanje brzine animacije, dodavanje navigacijskih gumba te prilagodbu izgleda. Sve te mogućnosti čine Swiper svestranim i prilagodljivim, a njegova lakoća korištenja, zajedno s jasnom dokumentacijom, čini ga preferiranim izborom za brzu integraciju i prilagodbu u različite projekte. Svojom lakoćom upotrebe i moćnim značajkama, Swiper se ističe kao izvrstan alat za stvaranje dinamičnih i vizualno atraktivnih klizača u modernom web razvoju.

4. Razvoj web aplikacije

Razvoj web aplikacije obuhvaća sveobuhvatan proces koji uključuje izradu korisničkog sučelja (engl. frontend) te pozadinskog dijela aplikacije (engl. backend). U ovom poglavlju istražiti će se osnovne komponente, procesi i tehnike koje se koriste pri implementaciji oba aspekata web aplikacije.

U prethodnom poglavlju proučene su, te odabrane tehnologije i alati koji će se koristiti kod razvoja web aplikacije. Prije samog početka potrebno je preuzeti i instalirati VSCode uređivač koda. Drugi preduvjet je instalacija Node.js-a koji predstavlja temelj za brojne alate i okvire koji se koriste u razvoju web aplikacija. Node.js omogućuje korištenje upravitelja paketa čvorova (engl. Node Package Manager, NPM) koji je nezamjenjiv alat u modernom web razvoju. NPM olakšava upravljanje bibliotekama i zavisnostima (engl. dependencies) potrebnim za projekt, omogućavajući brzu instalaciju, ažuriranje i integraciju raznih modula koji mogu značajno ubrzati i pojednostaviti razvojni proces.

Sve npm naredbe prilikom izrade ovog projekta pokretale su se pomoću integriranog terminala VSCode-a. Terminal, odnosno sučelje naredbenog retka (engl. Command-Line Interface, CLI) je tekstualni alat za unos koji omogućava korisnicima da komuniciraju s operativnim sustavom ili softverom putem naredbi unesenih u tekstualnom formatu. Terminal služi kao most između korisnika i računala, omogućujući izvođenje različitih zadataka kao što su upravljanje datotekama, pokretanje programa, te pristup i manipulaciju postavkama i resursima sustava. Za potrebe ovog projekta korištena je Node.js verzija 20.10.0, kodnog imena „Iron“, dok je verzija npm-a 10.2.3.

4.1. Frontend

Frontend se odnosi na dio web aplikacije ili web stranice s kojim korisnik dolazi u izravnu interakciju. To je „prednja strana“ aplikacije, vidljiva korisniku, i uključuje sve što korisnik može vidjeti i koristiti, poput korisničkog sučelja (engl. User Interface, UI) i korisničkog iskustva (engl. User Experience, UX). Frontend razvoj uključuje dizajn i implementaciju ovih vidljivih aspekata, osiguravajući da su funkcionalni, estetski privlačni i korisnički prihvatljivi.

Ključne tehnologije uključene u frontend razvoj su HTML, CSS i JavaScript.

HTML (engl. HyperText Markup Language) je standardni jezik za stvaranje i strukturiranje web stranica. HTML definira osnovnu strukturu i sadržaj web stranice.

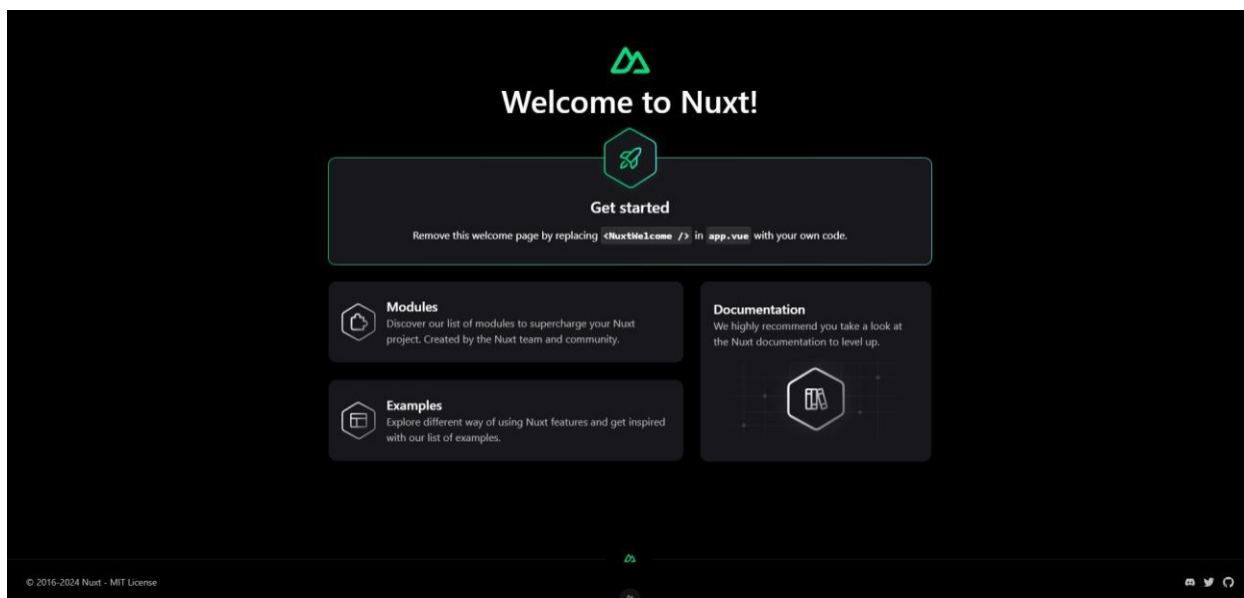
CSS (engl. Cascading Style Sheets) je stilski jezik koji se koristi za opis izgleda i formata dokumenata napisanih u HTML-u. CSS omogućava dizajnerima kontrolu nad izgledom web stranice, uključujući boje, fontove i raspored elemenata.

JavaScript je programski jezik koji omogućava dinamičko interaktivno ponašanje na web stranicama. JavaScript se koristi za dodavanje interaktivnih funkcija kao što su obrasci, animacije i manipulacija sadržajem na stranici.

Frontend razvoj zahtijeva razumijevanje kako se različiti web preglednici ponašaju te kako osigurati da web stranica ili aplikacija izgleda i funkcionira jednako dobro na svim uređajima i preglednicima. Također uključuje prilagodbu za različite veličine ekrana kroz responzivni dizajn, poboljšavajući pristupačnost i korisničko iskustvo za što širi raspon korisnika.

4.1.1. Kreiranje projekta

Razvoj frontend segmenta započinje kreiranjem novog Nuxt projekta, što se postiže naredbom „*npx nuxi@latest init optika-future*“. Jednom kada je projekt uspješno kreiran, sljedeći korak uključuje navigaciju do novostvorenog direktorija projekta korištenjem naredbe „*cd optika-future*“. Nakon toga, potrebno je inicirati instalaciju svih neophodnih paketa za Nuxt aplikaciju upotrebom naredbe „*npm install*“. Završetkom ovog procesa, projekt je spreman za pokretanje u razvojnom okruženju, što se postiže naredbom „*npm run dev*“. Ovom radnjom, Nuxt aplikacija postaje dostupna lokalno u pregledniku na poveznici <http://localhost:3000>. Slika 2 demonstrira automatski generiranu početnu stranicu Nuxt projekta, koja se može pregledati pristupanjem navedenom URL-u putem web preglednika.



Slika 2 Početna stranica novog Nuxt projekta

4.1.2. Instalacija i konfiguracija paketa

Uspješnim kreiranjem novog Nuxt projekta dolazimo na sljedeći korak koji uključuje instalaciju i konfiguraciju svih potrebnih alata, paketa i modula koji će se koristiti za razvoj web aplikacije.

Quasar

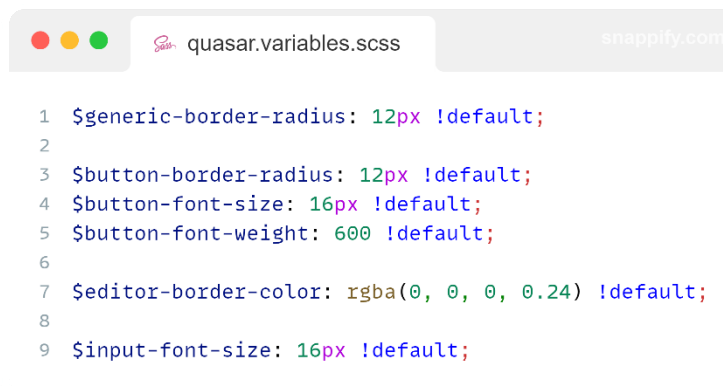
Instalacija Quasar modula pokreće se naredbom „*npm install quasar @quasar/extras nuxt-quasar-ui*“. Zatim slijedi konfiguracija samog modula. Slika 3 prikazuje *nuxt.config.ts* datoteku unutar koje se definira novoinstalirani Quasar modul kao i osnovna konfiguracija.

Kako bi dodatno prilagodili gotove UI komponente Quasar okvira, potrebno je stvoriti novu datoteka naziva *quasar.variables.scss* unutar *assets/css* direktorija. Unutar ove datoteke moguće je mijenjati predefinirane postavke stilova raznih komponenti definiranjem SCSS varijabli kako bi konačni dizajn aplikacije odgovarao željenom dizajnu. SCSS varijable koriste se za pohranu vrijednosti koje se obično ponovno koriste u CSS klasama, poput veličina fonta, boja, razmaka i više. Upotreba ovih varijabli zahtjeva instalaciju dodatnog paketa imena „SASS“, a instalira se naredbom „*npm install sass*“. Kompletna lista svih dostupnih varijabli može se proučiti na sljedećoj poveznici <https://quasar.dev/style/sass-scss-variables#variables-list>. Za potrebe ovog projekta promijenjeno je nekoliko zadanih postavki koje su prikazane na slici 4. Ovim korakom prilagođen je stil gotovih Quasar komponenti kako bi odgovarale dizajnu, odnosno stilu naše aplikacije.



```
1 export default defineNuxtConfig({
2   modules: ['nuxt-quasar-ui'],
3
4   quasar: {
5     config: {
6       brand: {
7         primary: '#224B8D',
8         secondary: '#E5E7EB',
9       },
10    },
11    sassVariables: 'assets/css/quasar.variables.scss',
12    extras: {
13      font: null,
14      fontIcons: ['fontawesome-v6', 'material-icons-outlined'],
15    },
16  },
17 });
```

Slika 3 Konfiguracija quasar modula unutar nuxt.config.ts datoteke



```
1 $generic-border-radius: 12px !default;
2
3 $button-border-radius: 12px !default;
4 $button-font-size: 16px !default;
5 $button-font-weight: 600 !default;
6
7 $editor-border-color: rgba(0, 0, 0, 0.24) !default;
8
9 $input-font-size: 16px !default;
```

Slika 4 quasar.variables.scss datoteka

Tailwind CSS

Dodavanje Tailwind CSS modula u projekt postiže se pokretanjem naredbe „*npm install -D @nuxtjs/tailwindcss*“. Oznaka *-D* (skraćeno od *--save-dev*) znači da će paket biti spremljen pod *devDependencies* unutar *packages.json* datoteke. Takvi paketi koriste se samo u razvojnoj fazi projekta, odnosno oni nisu potrebni za ispravno funkcioniranje aplikacije u fazi produkcije. Nakon instalacije modula, potrebno je dodati *@nuxtjs/tailwindcss* u sekciju *modules* unutar *nuxt.config.ts* datoteke. Konfiguracija Tailwind modula može se postići na dva načina. Prvi način sličan je kao i kod Quasar-a, definiranjem *tailwind* objekta unutar *nuxt.config.ts* datoteke. Drugi način, koji je ujedno i korišten u ovom projektu, je korištenjem zasebne konfiguracijske datoteke naziva *tailwind.config.js*. Pokretanjem naredbe „*npx tailwindcss init*“ stvara se navedena datoteka koja je prikazana na slici 5. Pošto Quasar i Tailwind sadrže

preklapanja u nazivima nekih klasa, kao što su na primjer klase *text-white* i *flex*, potrebno je definirati prefiks Tailwind klasa. Definicija prefiksa definirana je u retku 3 što je vidljivo na slici 5, a kao prefiks je odabran 'tw-'.

```
1  /** @type {import('tailwindcss').Config} */
2  module.exports = {
3    prefix: "tw-",
4    content: [],
5    theme: {
6      extend: {
7        colors: {
8          primary: "#224B8D",
9          accent: "#D4A418",
10         },
11        transitionProperty: {
12          font: "font-size",
13         },
14       },
15     },
16     plugins: [],
17   };
```

Slika 5 Konfiguracijska datoteka *tailwind.config.js*

Za definiranje globalnih klasa i stilova u Tailwind-u korištena je *tailwind.css* datoteka koja je smještena u istom direktoriju kao i datoteka za konfiguraciju Quasar SCSS varijabli (*assets/css* direktorij).

Slika 6 prikazuje primjer korištenja Tailwind „utility“ klasa. Ovaj primjer prikazuje implementaciju uvodne sekcije početne stranice web aplikacije.

```
1  <template>
2  <section class="tw-relative tw-h-svh tw-min-h-[600px]">
3    <q-img src="/images/women_eyeglasses.jpg" alt="Fotografija žene s naočalama" class="tw-h-full" />
4
5    <div class="tw-absolute tw-inset-0 tw-bg-primary/80">
6      <div class="tw-mx-auto tw-flex tw-h-full tw-max-w-4xl tw-flex-col tw-justify-center tw-gap-12 tw-px-6 lg:tw-max-w-5xl">
7        <h1 class="tw-text-pretty tw-text-center tw-text-4xl tw-font-bold tw-leading-normal tw-text-white md:tw-text-5xl lg:tw-text-7xl">
8          Otkrijte kristalno čisti vid
9        </h1>
10
11        <p class="tw-text-pretty tw-text-center tw-text-xl tw-text-white md:tw-text-xl lg:tw-text-2xl">
12          Rezervirajte termin za besplatan pregled vida već danas.
13        </p>
14
15        <div class="tw-text-center">
16          <q-btn no-caps color="white" text-color="primary" size="lg" label="Istraži više" />
17        </div>
18      </div>
19    </div>
20 </section>
21 </template>
```

Slika 6 Primjer korištenja *tailwind* "utility" klasa u implementaciji uvodne sekcije početne stranice

Swiper

Swiper se u Nuxt projekt dodaje instalacijom *nuxt-swiper* modula pokretanjem naredbe „*npm install nuxt-swiper*“. Nakon instalacije potrebno je dodati *nuxt-swiper* u sekciju *modules* unutar *nuxt.config.ts* datoteke. Ovaj modul također ima mogućnost definiranja dodatne konfiguracije, no za potrebe ovog projekta to nije bilo potrebno.

Primjer implementacije Swiper paketa detaljno je ilustriran na slici 7, kroz primjenu u komponenti *ProductCarousel.vue*. Swiper arhitektura se oslanja na dva ključna elementa: *Swiper*, koji služi kao omot (engl. *wrapper*) za slajdove, te *SwiperSlide*, koji predstavlja pojedinačne slajdove unutar omota. Konfiguracija Swiper paketa odvija se u *<script>* dijelu *.vue* datoteke, gdje se objektima *setings* i *breakpoints* definiraju osnovne i adaptivne postavke Swiper-a. Početne postavke se navode unutar *setings* objekta, dok *breakpoints* objekt omogućava prilagodbu postavki baziranih na dimenzijama zaslona, optimizirajući tako prikaz za različite uređaje. Dodatne funkcionalnosti, poput paginacije, integriraju se putem *modules* atributa, gdje se specificira željeni modul, u ovom slučaju modul za paginaciju. Lokacija paginacije se prilagođava koristeći opciju *el*, omogućavajući time fleksibilnost u pozicioniranju elementa paginacije. Interaktivnost s paginacijom postiže se putem *clickable* atributa, koji omogućava navigaciju klikom na elemente paginacije. *SlidesPerView* atribut određuje koliko slajdova se prikazuje u jednom trenutku, s adaptivnim prilagodbama unutar *breakpoints* objekta za različite veličine zaslona, dok *spaceBetween* atribut kontrolira razmak između slajdova. Za slučajeve kada je dostupan broj slajdova manji od definiranog broja za prikaz, *centerInsufficientSlides* opcija osigurava centriranje slajdova. Standardno, Swiper omogućava navigaciju klizanjem, ali za navigaciju putem gumbi koristi se *SwiperNavigation* modul, ili, alternativno, prilagodba putem Quasar-ovih *q-btn-group* i *q-btn* komponenti za stvaranje prilagođene navigacije kako je prikazano u ovom primjeru. Koristimo referencu na *Swiper* komponentu kako bi smo pristupili *slideNext()* i *slidePrev()* funkcijama za kretanje među slajdovima. Osim toga, atributi *isBeginning* i *isEnd* služe kao indikatori za deaktivaciju navigacijskih gumbi kada se korisnik nalazi na početnom ili završnom slajdu, dodatno poboljšavajući korisničko iskustvo kroz intuitivnu interakciju.

```

1 <script lang="ts" setup>
2 defineProps<{
3   products: Product[];
4   title: string;
5 }>();
6
7 const settings = {
8   modules: [SwiperPagination],
9   pagination: {
10    el: ".offers__pagination",
11    clickable: true,
12  },
13  slidesPerView: 1,
14  spaceBetween: 24,
15  centerInsufficientSlides: true,
16 };
17 const breakpoints = {
18   "460": {
19     slidesPerView: 2,
20   },
21   "768": {
22     slidesPerView: 3,
23   },
24   "1024": {
25     slidesPerView: 4,
26   },
27 };
28
29 const swiperRef = ref();
30 const getRef = (swiper: any) => {
31   swiperRef.value = swiper;
32 };
33 </script>
34
35 <template>
36   <section class="...">
37     <div...
38     </div>
39
40     <div class="tw-min-h-72">
41       <Swiper
42         v-bind="settings"
43         :breakpoints="breakpoints"
44         class="!tw-p-6"
45         @swiper="getRef"
46       >
47         <SwiperSlide
48           v-for="product in products"
49           :key="product.id"
50           class="!tw-h-auto"
51         >
52           <CardsProduct :product="product" />
53         </SwiperSlide>
54       </Swiper>
55     </div>
56
57     <div class="...">
58       <div class="...">
59         <div class="offers__pagination"></div>
60       </div>
61
62       <q-btn-group flat class="...">
63         <q-btn ...
64           :disable="swiperRef?.isBeginning"
65           @click="swiperRef?.slidePrev()"
66         />
67         <q-btn ...
68           :disable="swiperRef?.isEnd"
69           @click="swiperRef?.slideNext()"
70         />
71       </q-btn-group>
72     </div>
73   </section>
74 </template>

```

Slika 7 Primjer korištenja Swiper biblioteke

Vuefire

S obzirom da je za potrebe konfiguracije vuefire modula potreban Firebase projekt, instalacija i konfiguracija modula biti će opisana u poglavlju 4.2.2. Povezivanje Firebase projekta s web aplikacijom.

Google Maps

Ovaj projekt koristit će Google karte za prikaz lokacija poslovnica. Za jednostavno korištenje i upravljanje Google kartama u Nuxt projektu koristit će se *vue3-google-map* paket. Instalacija paketa postiže se naredbom „*npm install vue3-google-map*“. Da bi se karte ispravno prikazivale potrebno je definirati *Google Maps API key* unutar *.env* datoteke. Ova datoteka sadrži varijable okruženja (engl. environment variables), a to su globalne varijable koje sadrže vrijednosti kao što su ključevi (engl. keys) i razni konfiguracijski podaci koji se dohvaćaju od strane aplikacije po potrebi.

Slika 8 prikazuje primjer korištenja *vue3-google-map* biblioteke kako je implementirano u ovom projektu.

U *script* dijelu *vue* datoteke učitane su komponente *GoogleMap*, *Marker* i *InfoWindow*. Definiran je konfiguracijski objekt koji izvlači potrebne postavke za vrijeme izvođenja, kao što je API ključ. Referenca karte *mapRef* i niz *branches* koriste se za spremanje informacija o karti i lokacijama poslovnica.

Funkcija *getCenterWithOffset* je kreirana kako bi se prilagodila centralna točka karte ovisno o širini prozora preglednika. Ova funkcija vraća objekt s geografskim koordinatama koji se koristi za određivanje središnje točke karte. Središnja točka *center* je reaktivna referenca koja se postavlja na prvu lokaciju iz niza *branches*. Funkcija *onLocationChange* aktivira se kada korisnik odabere lokaciju. Ona pretražuje niz *branches* kako bi pronašla novu centralnu točku karte i zatim ažurira trenutni prikaz pomoću metode *setCenter*.

U *template* dijelu, korištena je *GoogleMap* komponenta s definiranim početnim postavkama poput središnje točke i razine zumiranja. Opcijom *disable-default-ui* isključili smo zadane navigacijske elemente koji dolaze integrirani s Google kartom. Za svaku poslovnicu unutar niza *branches*, prikazan je *Marker* koji sadrži *InfoWindow* za prikaz imena poslovnice, omogućujući korisnicima da dobiju više informacija klikom na marker.

```
BranchLocations.vue
1 <script lang="ts" setup>
2 import { GoogleMap, Marker, InfoWindow } from "vue3-google-map";
3
4 const config = useRuntimeConfig();
5 const mapRef = ref();
6 const branches = [...];
7
8 const getCenterWithOffset = (center: { lat: number; lng: number }) => {
9   if (window.innerWidth < 768) {
10     return { lat: center.lat + 0.002, lng: center.lng };
11   }
12   return { lat: center.lat, lng: center.lng - 0.005 };
13 };
14
15 const center = ref(getCenterWithOffset(branches[0].center));
16
17 const onLocationChange = (e: any) => {
18   const newCenter = branches.find(
19     (branch) => branch.name === e.srcElement.innerText,
20   );
21   if (newCenter) {
22     mapRef.value.map.setCenter(getCenterWithOffset(newCenter.center));
23   }
24 };
25 </script>
26
27 <template>
28 <div class="tw-relative">
29   <div ...
30     </div>
31
32   <GoogleMap
33     ref="mapRef"
34     :api-key="config.public.mapsApiKey"
35     class="tw-h-[600px] tw-w-full"
36     :center="center"
37     :zoom="16"
38     :disable-default-ui="true"
39   >
40     <Marker
41       v-for="marker in branches"
42       :key="marker.id"
43       :options="{ position: marker.center }"
44     >
45       <InfoWindow
46         :options="{ position: marker.center, content: marker.name }"
47       />
48     </Marker>
49   </GoogleMap>
50 </div>
51 </template>
```

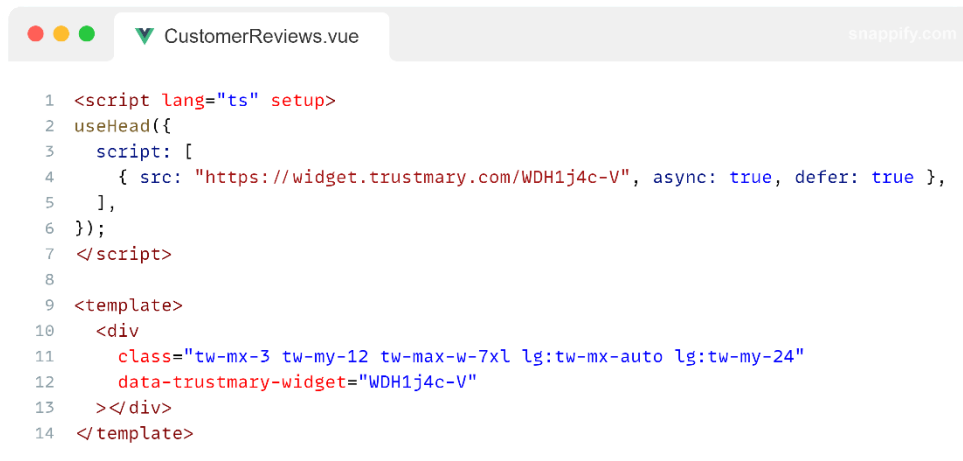
Slika 8 Primjer korištenja *vue3-google-map* biblioteke

Trustmary

Za integraciju recenzija korisnika na web stranicu prvo je potrebno postaviti temelje stvaranjem Trustmary računa. Ovaj korak otvara vrata personalizaciji, omogućujući da oblikujemo novi *widget* koji se savršeno uklapa u estetiku naše web aplikacije. Prilagoditi se može stil, tema, raspored i jezik *widget-a*, uz mogućnost detaljnijeg finog podešavanja putem prilagođenog CSS-a za one koji žele dodatno uskladiti izgled prikaza recenzija, odnosno *widget-a*.

Kada je *widget* spreman, sljedeći korak je popunjavanje recenzijama. Trustmary nudi fleksibilnost u izboru izvora recenzija, uključujući popularne platforme poput Google-a, Facebook-a, TripAdvisor-a i Yelp-a, kao i opciju za ručni unos podataka iz Excela ili CSV datoteka za maksimalnu kontrolu nad sadržajem.

Da bismo recenzije učinili dijelom naše web aplikacije, upotrebljava se *useHead()* funkcija, koja nudi jednostavan i reaktivan način za upravljanje HTML *head* elementom. Primjer integracije Trustmary recenzija u web aplikaciju prikazan je na slici 9, demonstrirajući kako se ovaj alat može koristiti za obogaćivanje korisničkog iskustva kroz stvarne ocjene kupaca. Dodatno, bitno je napomenuti da je komponentu koja definira Trustmary widget potrebno omotati (engl. wrap) ClientOnly komponentom kako bi se ona prikazivala samo na klijentskoj strani, odnosno u pregledniku korisnika. U suprotnom se Trustmary widget prikaže dva puta, jednom na serveru (ako se koristi SSR), te jednom u pregledniku.



```
1 <script lang="ts" setup>
2 useHead({
3   script: [
4     { src: "https://widget.trustmary.com/WDH1j4c-V", async: true, defer: true },
5   ],
6 });
7 </script>
8
9 <template>
10   <div
11     class="tw-mx-3 tw-my-12 tw-max-w-7xl lg:tw-mx-auto lg:tw-my-24"
12     data-trustmary-widget="WDH1j4c-V"
13   ></div>
14 </template>
```

Slika 9 Primjer dodavanja Trustmary widget-a u Nuxt aplikaciju

TypeScript, ESLint i Prettier

Iako svaki od ovih alata možemo promatrati kao zasebnu cjelinu, ukoliko se koriste u istom projektu potrebno je na ispravan način konfigurirati sva tri alata kako ne bi došlo do potencijalnih problema. Krenimo sa instalacijom TypeScript paketa naredbom „*npm install -D typescript*“. Pravila za kompilaciju TypeScript koda definiraju se u *tsconfig.json* datoteci. Ova datoteka dolazi predefinirana instalacijom Nuxt projekta jer Nuxt u pozadini koristi TypeScript stoga nije potreba daljnja konfiguracija, no ovisno o projektu moguće je definirati dodatna pravila.

Slijedi instalacija ESLint-a i pomoćnih alata naredbom „*npm install -D eslint @typescript-eslint/parser @nuxtjs/eslint-config-typescript*“. Koristeći *@typescript-eslint/parser* kao parser u ESLint konfiguraciji

omogućava se da ESLint analizira TypeScript kod. „Parser je dio kompajlera koji rastavlja izvorni kod u male dijelove nizova znakova koji se nazivaju tokeni.“ (Rouse, 2022). Paket `@nuxtjs/eslint-config-typescript` sadrži konfiguraciju koja definira pravila specifična za TypeScript, osiguravajući da ESLint i TypeScript mogu zajedno funkcionirati.

Sada je potrebno instalirati Prettier naredbom „`npm install -D prettier prettier-plugin-tailwindcss eslint-config-prettier eslint-plugin-prettier`“. Dodatak `prettier-plugin-tailwindcss` koristi se za automatsko sortiranje Tailwind klasa kako bi se poboljšala čitljivost i konzistentnost koda. Da bi ovaj dodatak funkcionirao potrebno ga je definirati u `.prettierrc` konfiguracijskog datoteci kako je prikazano na slici 10.

```

1 {
2   "plugins": ["prettier-plugin-tailwindcss"]
3 }

```

Slika 10 Prettier konfiguracijska datoteka `.prettierrc`

Da bi se izbjegli konflikti između Prettier-a i ESLint-a, koristi se `eslint-config-prettier` koji isključuje sva ESLint pravila koja bi mogla doći u sukob s Prettier-om. Također, `eslint-plugin-prettier` omogućava pokretanje Prettier-a kao ESLint pravila, što osigurava da se svaki kod automatski formatira prije nego što prođe ESLint provjeru.

U tablici 1 prikazana je dodatna usporedba između ESLint-a i Prettier-a kako bi se što bolje shvatila njihova svrha, namjena te razlike među njima.

Tablica 1 Usporedba ESLint-a i Prettier-a (Ccari 2023)

	ESLint	Prettier
Priroda	Alat za statičku analizu koda i linter za JavaScript	Alat za oblikovanje koda, koji je primarno usmjeren na oblikovanje koda kako bi isti bio čitljiviji i dosljedniji
Svrha	Kvaliteta koda i otkrivanje grešaka	Oblikovanje koda
Konfiguracija	Visoko prilagodljiv, pravila se mogu definirati u <code>.eslintrc</code> konfiguracijskoj datoteci	Manje prilagodljiv jer provodi određeni skup pravila bez konfiguracije, no još uvijek postoji opcija konfiguracije nekih opcija u <code>.prettierrc</code> datoteci
Integracija	Može se integrirati s popularnim IDE-ovima kako bi pružio povratne informacije u stvarnom vremenu i mogućnosti automatskog popravljaja	Može se integrirati s popularnim IDE-ovima kako bi se datoteke formatirale prilikom spremanja kako bi se ispravili problemi s formatiranjem

Za kraj je preostalo konfigurirati ESLint unutar `.eslintrc.js` datoteke kako je prikazano na slici 11. Primijetimo kako je u liniji 14 dodatno definirano pravilo koje isključuje zadano pravilo Vue.js-a koje govori da se nazivi Vue komponenata moraju sastojati od više riječi. To pravilo sprječava sukobe s postojećim HTML elementima, jer su nazivi svih HTML elemenata sastojeci od jedne riječi, no kako će se u ovom projektu koristiti komponente s nazivima sastavljenim od jedne riječi (npr. Product, Banner) odlučeno je isključiti to pravilo.



```
1 module.exports = {
2   root: true,
3   env: {
4     browser: true,
5     node: true,
6   },
7   parser: 'vue-eslint-parser',
8   parserOptions: {
9     parser: '@typescript-eslint/parser',
10  },
11  extends: ['@nuxtjs/eslint-config-typescript', 'plugin:prettier/recommended'],
12  plugins: [],
13  rules: {
14    'vue/multi-word-component-names': 'off',
15  },
16  };
```

Slika 11 Konfiguracija ESLinta u .eslintrc.js datoteci

4.2. Backend

Backend se odnosi na poslužiteljsku stranu aplikacije, koja je nevidljiva korisnicima, ali ključna za funkcionalnost i upravljanje procesima. Ovaj dio aplikacije uključuje poslužitelje, baze podataka, i poslužiteljske aplikacije koje rade u pozadini kako bi obradile zahtjeve, upravljale korisničkim podacima, autentifikacijom, autorizacijom, i ostalim poslovnim logikama koje omogućavaju aplikaciji da funkcioniра.

Backend razvoj uključuje programiranje poslužiteljskih skripti, konfiguriranje baza podataka, osiguravanje sigurnosti aplikacije, te kreiranje aplikacijskog programskog sučelja (engl. Application Programming Interface, API) koji omogućuje komunikaciju između klijentske (frontend) i poslužiteljske (backend) strane. Tehnologije često korištene u backend razvoju uključuju različite programerske jezike poput JavaScript-a (Node.js), Python-a (Django, Flask), Ruby-a (Ruby on Rails), Java (Spring), te baze podataka kao što su MySQL, PostgreSQL, MongoDB i druge. Učinkovit backend je temelj za stabilne, sigurne i skalabilne web aplikacije, sposobne za obradu velikog broja zahtjeva i podataka bez kompromitiranja performansi ili korisničkog iskustva.

U slučaju ovog projekta korišten je Firebase kao backend rješenje pomoću kojeg razvoj dobiva jedinstveni pristup koji omogućava brz i efikasan razvoj aplikacije bez potrebe za eksplicitnim upravljanjem serverima ili infrastrukturom.

4.2.1. Kreiranje projekta

Razvoj backend-a započeo je kreiranjem novog Firebase projekta u Firebase konzoli. Prvo se unosi ime aplikacije, odnosno naziv projekta. Zatim je ponuđena opcija uključivanja Google analitike (engl. Google Analytics) za novo kreirani projekt. Google analitika je besplatna usluga za praćenje i izvještavanje prometa na web stranici. Ona omogućuje korisnicima detaljnu analizu posjećenosti web stranica, uključujući izvore

prometa i ponašanje korisnika, što pomaže u optimizaciji web sadržaja i marketinških strategija. Zbog prethodno spomenutih prednosti odlučeno je koristiti ovu opciju koja će u budućnosti pomoći kod poboljšanja web aplikacije. Zadnji korak kreiranja projekta je odabir računa putem kojeg će se omogućiti pristup Google analitici.

4.2.2. Povezivanje Firebase projekta s web aplikacijom

Nakon uspješno kreiranog projekta potrebno je povezati Firebase s web aplikacijom. Na početnoj stranici Firebase konzole odabire se opcija dodavanja web aplikacije. Prvo je potrebno registrirati aplikaciju unošenjem nadimka aplikacije te odabirom opcije „Register app“. Prije samo registracije moguće je dodati i uslugu Firebase Hosting-a što je odmah i učinjeno pošto će se ta usluga koristit kasnije prilikom razvoja aplikacije.

Uspješnom registracijom dolazi se na sljedeći korak, a to je instalacija Firebase paketa za razvoj programa (engl. Software Development Kit, SDK) te inicijalizacija Firebase-a. Radi lakše integracije Firebase-a sa Nuxt razvojnim okvirom odlučeno je korištenje dodatnog modula naziva Vuedfire. Instalacija potrebnih paketa pokreće se naredbom „*npm install vuedfire nuxt-vuedfire firebase*“. Nakon uspješne instalacije paketa potrebno je postaviti konfiguraciju *vuedfire* paketa kako bi web aplikacija mogla komunicirati s prethodno kreiranim Firebase projektom. Konfiguracija se nalazi unutar *nuxt.config.ts* datoteke, a prikazana je na slici 12. U drugoj liniji definiramo prethodno instalirani *nuxt-vuedfire* modul, a zatim slijedi *vuedfire* konfiguracija. S obzirom da ćemo koristiti Firebase autentifikaciju potrebno je omogućiti (engl. enable) *auth* opciju. Zatim slijedi *config* objekt unutar kojeg se definira konfiguracija koju smo dobili prilikom registracije aplikacije u prethodnom odlomku. U ovom slučaju konfiguracijski podaci definirani su u *.env* datoteci, a unutar *nuxt.config.ts* datoteke može im se pristupiti korištenjem *process.env* sintakse.



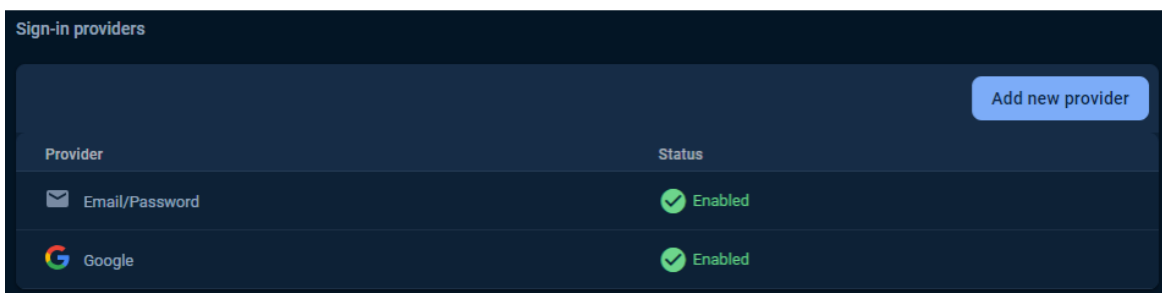
```
1 export default defineNuxtConfig({
2   modules: ["nuxt-vuedfire"],
3
4   vuefire: {
5     auth: {
6       enabled: true,
7     },
8     config: {
9       apiKey: process.env.FB_API_KEY,
10      authDomain: process.env.FB_AUTH_DOMAIN,
11      projectId: process.env.FB_PROJECT_ID,
12      appId: process.env.FB_APP_ID,
13      storageBucket: process.env.FB_STORAGE_BUCKET,
14      messagingSenderId: process.env.FB_MESSAGING_SENDER_ID,
15      measurementId: process.env.FB_MEASUREMENT_ID,
16    },
17  },
18 });
```

Slika 12 Konfiguracija vuedfire paketa unutar nuxt.config.ts datoteke

4.2.3. Postavljanje Firebase autentifikacije

Postavljanje Firebase autentifikacije započinje sa odabirom opcije „Authentication“ koja se nalazi u bočnom izborniku Firebase konzole unutar „Build“ skupine. Zatim, klikom na „Get started“, prikazuje se nadzorna ploča autentifikacije koja se sastoji od sljedećih kartica: Users, Sign-in method, Templates, Usage, Settings i Extensions.

Novi projekti nemaju aktiviranu niti jednu opciju za prijavu, stoga je prvo potrebno odabrati karticu Sign-in method kako bi postavili novu opciju prijave. Za potrebe ovog projekta odabrane su dvije opcije. Prva odabrana opcija, Email/Password, omogućuje prijavu i registraciju korisnika unošenjem emaila i lozinke. Druga odabrana opcija je Google provider koja omogućuje jednostavnu i sigurnu prijavu korisnika putem njihovih Google računa u web aplikaciju. Slika 13 prikazuje popis odabranih opcija prijave u Firebase konzoli.



Slika 13 Popis odabranih opcija prijave u Firebase konzoli

4.2.4. Postavljanje Firestore baze podataka

Baza podataka postavlja se odabirom Firestore Database opcije iz bočnog izbornika, a zatim klikom na „Create database“ kreira se nova baza. Kod kreiranje baze, prvo je potrebno odabrati lokaciju. Odabire se ona lokacija najbliža vašim korisnicima kako bi brzine preuzimanja sadržaja bile što veće. U slučaju ovog projekta odabrana je lokacija „eur3 (Europe)“. Nakon ovog koraka odabrana se opcija „Start in test mode“, koja dopušta sve radnje čitanja i pisanja unutar baze podataka sljedećih 30 dana. Ova opcija koristi se samo za vrijeme razvoja web aplikacije, no prije nego se aplikacija pusti u produkciju potrebno je definirati sigurnosna pravila tako da samo ovlašteni korisnici imaju pravo na uređivanje i čitanje podataka.

4.2.5. Postavljanje Firebase skladišta podataka

Kao i u prethodna dva slučaja, započinje se odabirom „Storage“ u bočnom izborniku te klikom na „Get started“ stvara se novo skladište podataka. Ovdje je također odabrana opcija „Start in test mode“ koja se koristi samo za vrijeme razvoja web aplikacije. Lokacija se automatski postavlja prema lokaciji odabranoj prilikom postavljanja baze podataka.

4.2.6. Postavljanje Firebase poslužitelja

Postavljanje se započinje odabirom „Hosting“ opcije iz bočnog izbornika te klikom na „Get started“. Prvo je potrebno instalirati Firebase alat naredbenog retka (engl. Command Line Interface, CLI), a to se postiže naredbom „*npm install -g firebase-tools*“. Oznaka *-g* (skraćeno od *--global*) govori NPM-u da paket instalira globalno, što znači da će taj paket biti dostupan u svim projektima na operativnom sustavu, za razliku od lokalnih paketa koji su dostupni samo unutar projekata u kojima su instalirani.

Nakon instalacije Firebase CLI, potrebno je pokrenuti naredbu *firebase login*, a zatim se prijaviti računom putem kojeg je kreiran projekt. Prijava omogućuje povezivanje Firebase projekta sa lokalnim projektom web aplikacije. Sada je na redu inicijalizacija poslužitelja, a to se postiže komandom *firebase init hosting*. Pošto Firebase i lokalni projekt još uvijek nisu povezani potrebno je odabrati opciju „Use an existing project“, te iz ponuđene liste odabrati željeni projekt. U ovom slučaju odabran je projekt *optika-future*. Slijedi niz pitanja za konfiguraciju poslužitelja. Kao javni direktorij odabrao se *.output/public* iz razloga što je to zadani direktorij unutar kojeg se izgrađuje Nuxt aplikacija. Sve ostale opcije ostavljene su kao zadane vrijednosti.

5. Pregled funkcionalnosti web aplikacije

U ovom dijelu rada detaljno će se razmotriti ključne funkcionalnosti web aplikacije, s posebnim osvrtom na različite razine pristupa i mogućnosti koje su pridružene svakoj korisničkoj ulozi. Specifično, razlikovat ćemo između gostiju (korisnika koji nisu registrirani), klijenata (korisnika koji su prošli proces registracije) i administratora. Za svaku od ovih kategorija, opisat ćemo njihove jedinstvene mogućnosti i ograničenja unutar aplikacije, pružajući uvid u način na koji su servisi i informacije strukturirani i dostupni različitim korisnicima. Analiza će obuhvatiti i interakciju između ovih uloga te kako one doprinose cjelokupnom korisničkom iskustvu i upravljanju web platformom.

Gosti, kao neregistrirani korisnici, mogu pristupiti samo osnovnim funkcionalnostima web aplikacije koje ne zahtijevaju autentifikaciju. Ovo obično uključuje pregled sadržaja, upoznavanje s ponudom, korištenje kontakt obrazaca i pregled osnovnih informacija o tvrtki, kao na primjer adrese, kontakt brojevi i email.

Klijenti, ili registrirani korisnici, imaju proširene mogućnosti koje uključuju interakcije poput naručivanja proizvoda ili usluga, pristup personaliziranom sadržaju i mogućnost sudjelovanja u programima vjernosti. Osim toga, klijenti mogu upravljati svojim osobnim podacima, pregledom povijesti kupnji i praćenjem statusa narudžbi.

Administratori su, pak, korisnici s najvišim razinama pristupa i ovlastima unutar web aplikacije. Oni imaju kapacitet za upravljanje sadržajem, korisničkim računima, procesiranjem narudžbi, te imaju pristup analitičkim podacima i alatima za izvještavanje. Uloga administratora je centralna za održavanje i ažuriranje web aplikacije, te za implementaciju promjena koje mogu utjecati na iskustvo i zadovoljstvo svih korisnika.

Ovaj pregled pruža temelj za razumijevanje kako se web aplikacija prilagođava potrebama i privilegijama različitih korisnika, osiguravajući da svaki korisnik ima pristup funkcionalnostima koje su relevantne za njegovu ulogu.

5.1. Gosti (neregistrirani korisnici)

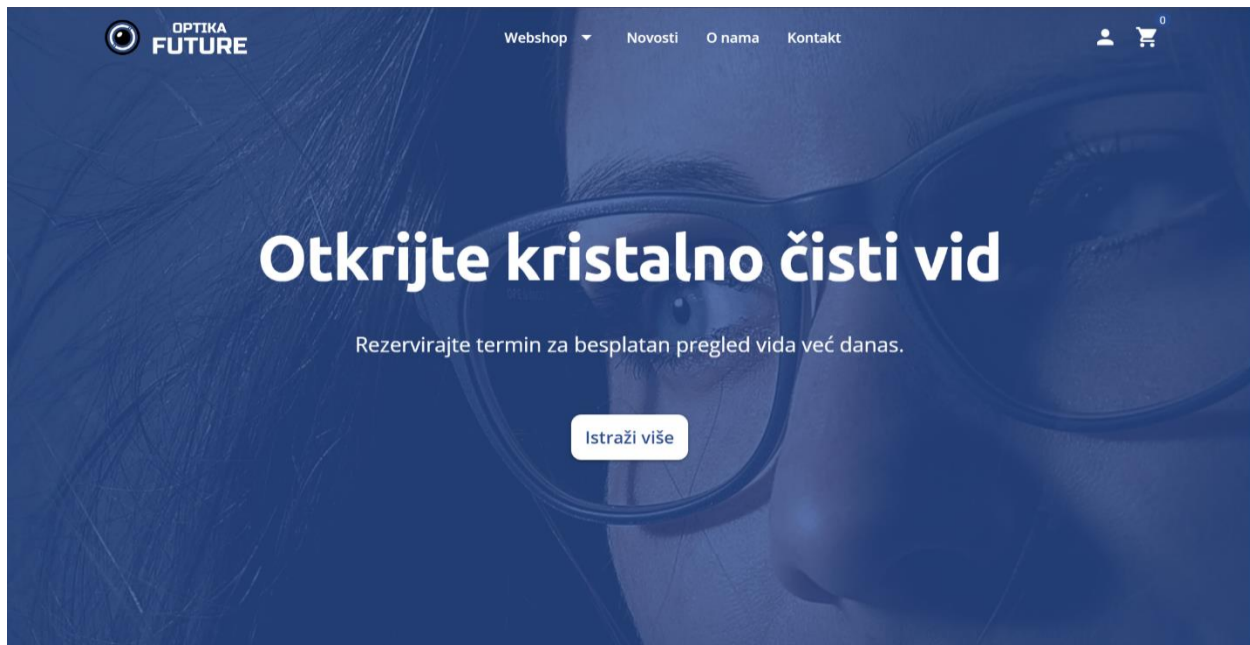
5.1.1. Početna stranica

Prvi dojam je često presudan, a početna stranica web aplikacije „Optika Future“ oblikovana je s ciljem stvaranja jasnog i privlačnog korisničkog dojma. Ključni elementi koje korisnik prvo primjećuje jesu poruka i poziv na akciju (engl. Call To Action, CTA).

„Call to Action je poruka koja usmjerava korisnike na sljedeći logičan korak u prodajnom procesu. CTA je obično riječ ili kratka fraza kojom uvjeravamo osobu da mora djelovati u onom smjeru kojem mi kao prodavač želimo. Ponekad želimo da osoba počne koristiti našu uslugu, da nas nazove, da komentira članak, da pogleda neki drugi članak, da se pretplati na newsletter ili pak da kupi nešto.“ (Jagodić 2022).

Glavni vizual, koji dominira početnom stranicom (slika 14), prikazuje jasno i fokusirano oko kroz naočale, što metaforički naglašava glavnu uslugu – poboljšanje vida. Upotreba tamnoplave boje u pozadini stvara osjećaj profesionalnosti i pouzdanosti, dok bijela tipografija osiguravaju dobru čitljivost i kontrast. Poruka „Otkrijte kristalno čisti vid“ ne samo da informira korisnike o glavnoj vrijednosti koju nudi „Optika Future“, već i poziva na akciju s ponudom besplatnog pregleda vida. Poziv na akciju, „Istraži više“, direktno komunicira s korisnicima nudeći im mogućnost pregleda i rezervacije termina.

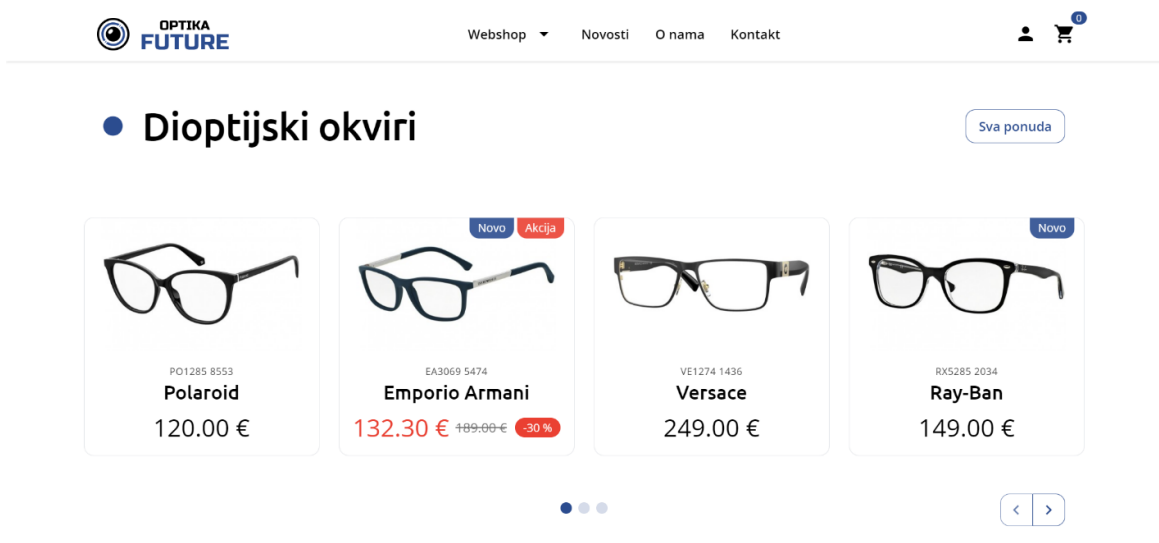
Ova početna stranica je optimizirana kako bi odmah privukla korisnika jasnim prikazom vrijednosti i usluga, te pružila jednostavan put do daljnjih akcija. Također, dizajn je prilagođen kako bi bio odzivan na različitim uređajima, osiguravajući da je prvi dojam pozitivan neovisno o tome pristupa li se stranici s mobilnog telefona, tableta ili desktop računala.



Slika 14 Početna stranica web aplikacije "Optika Future"

Navigacijska traka je smještena na vrhu stranice, nudeći jednostavan i intuitivan pristup različitim dijelovima web stranice: Webshop, Novosti, O nama, i Kontakt. Ovo omogućava korisnicima lako snalaženje i pristup informacijama koje ih zanimaju bez preopterećenja ili zbunjujućih elemenata.

Kako korisnici nastavljaju svoje putovanje kroz početnu stranicu web aplikacije „Optika Future“, nakon uvodnog segmenta poziva na akciju, dolazimo do prikaza proizvoda što je vidljivo na slici 15. Prikaz izdvojenih proizvoda je intuitivno organiziran u obliku karusela, omogućavajući korisnicima da jednostavnim klizanjem pregledaju različite modele. Svaki artikl je jasno prikazan s fotografijom, imenom proizvođača, modelom i cijenom, uz dodatne oznake poput „Novo“ ili „Akcija“ koje privlače pažnju i informiraju kupce o trenutnim ponudama. Opcija „Sva ponuda“ nudi zainteresiranim kupcima jednostavan način da otkriju cjelokupni asortiman proizvoda.



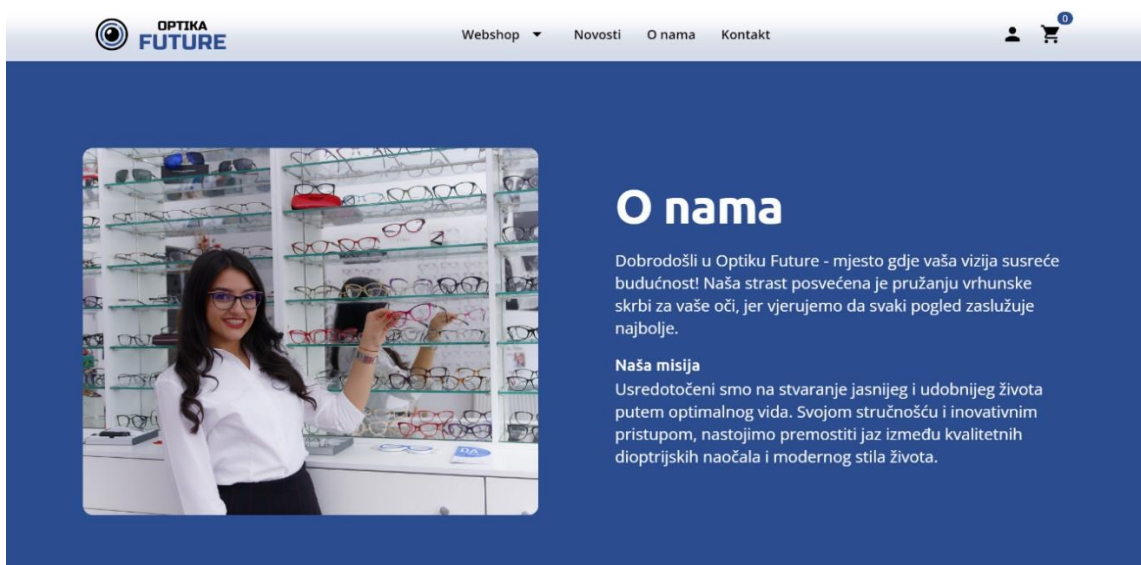
Slika 15 Karusel izdvojene ponude na početnoj stranici

Karusel se prilagođava veličini zaslona pa ukoliko korisnik pristupa web aplikaciji putem mobilnog uređaja, tada će mu biti prikazan samo jedan proizvod s mogućnošću navigacije kako je prikazano na slici 16.



Slika 16 Karusel izdvojene ponude na mobilnim uređajima

Daljnijm pomicanjem niz početnu stranicu, posjetitelji nailaze na sekciju „O nama“ koja je prikazana na slici 16. Ova sekcija istaknuta je primarnom plavom bojom s bijelom tipografijom. Sekcija „O nama“ dostupna je i izravno putem navigacijske trake.



Slika 17 Sekcija "O nama" na početnoj stranici

Nakon sekcije „O nama“ slijedi još jedna sekcija s pregledom izdvojene ponude, no ovog puta sunčanih naočala. Ova sekcija koristi istu komponentu za prikaz proizvoda kao i prošla stoga je nećemo detaljno opisivati.

Zatim slijedi sekcija „Kontakt“ (slika 18) koja je ključna za uspostavljanje komunikacije s klijentima. Ova sekcija dizajnirana je da bude jednostavna i intuitivna, olakšavajući korisnicima da brzo i lako stupaju u kontakt s tvrtkom. Kao i kod sekcije „O nama“ i ova sekcija izravno je dostupna putem navigacijske trake.

Za one koji preferiraju osobnu komunikaciju, poveznica "Pronađi poslovnice" vodi do informacija o fizičkim lokacijama, omogućavajući korisnicima da brzo pronađu najbližu poslovnicu.

Slika 18 Sekcija "Kontakt" na početnoj stranici

Ispunjavanjem kontakt obrasca i slanjem upita, prikazuje se poruka zahvalnosti kao što je prikazano na slici 19. Ova poruka bitan je dio korisničkog iskustva jer potvrđuje uspješno slanje poruke i osigurava korisnike da je njihov upit primljen i da će biti obrađen. Dodatno, opcija „Pošalji novu poruku“ omogućava korisnicima slanje nove poruke bez potrebe za osvježavanjem stranice.



Slika 19 Prikaz poruke zahvalnosti nakon slanja upita

Poslije sekcije „Kontakt“ dolazimo do sekcije „Novosti“ koja je prikazana na slici 20. Ova sekcija služi kao središnje mjesto za najnovije informacije, savjete i obavijesti koji se odnose na proizvode i usluge koje nudi optika. Kroz pažljivo odabrane članke i ažuriranja, „Novosti“ pružaju vrijedan sadržaj kako postojećim tako i potencijalnim korisnicima. Svaki članak prikazan je slikom, datumom objave, naslovom, kratkim uvodom te tagovima. Cijeli članak prikazuje se klikom na bilo koje područje objave što će biti jasno naznačeno zumiranjem slike i promjene boje naslova u primarnu plavu boju. Opcija „Istraži više“ vodi korisnika na pregled svih članaka.

● Novosti

[Istraži više](#)

18. 02. 2024.

Vodič kroz trendove naočala za 2024: Budite u toku s...

Kako se približavamo 2024., svijet naočala doživljava revoluciju ne samo u funkcionalnosti, već i u modnom izričaju. Od vintage okvira koji odišu nostalgijom...

[Trendovi](#) [2024](#)

26. 01. 2024.

Fotoosjetljive leće

Jeste li ikada poželjeli da vaše naočale mogu čitati vaš um? Zamislite leće koje se automatski prilagođavaju svjetlini sunca, štiteći vaše oči i pružajući vam kristal...

[Fotoosjetljive leće](#) [UV zrake](#)

10. 01. 2024.

Zaštita očiju od digitalnog napreznja

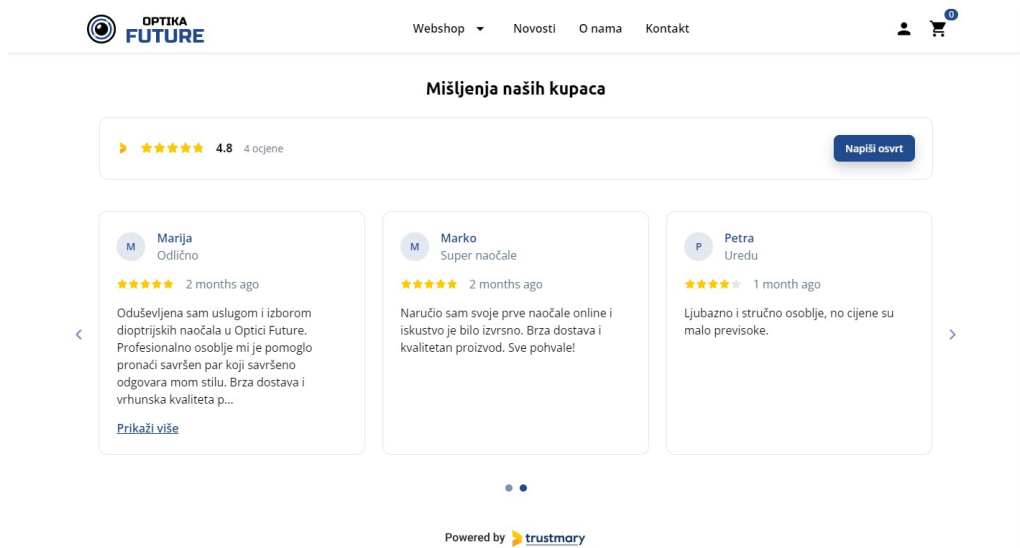
Ah, digitalni svijet! Ne možemo živjeti s njim, ne možemo živjeti bez njega. Dok surfamo valovima interneta, gledamo omiljene serije ili radimo od kuće, n...

[zaštita](#) [plavo svjetlo](#) [ekran](#)

Slika 20 Sekcija "Novosti" na početnoj stranici

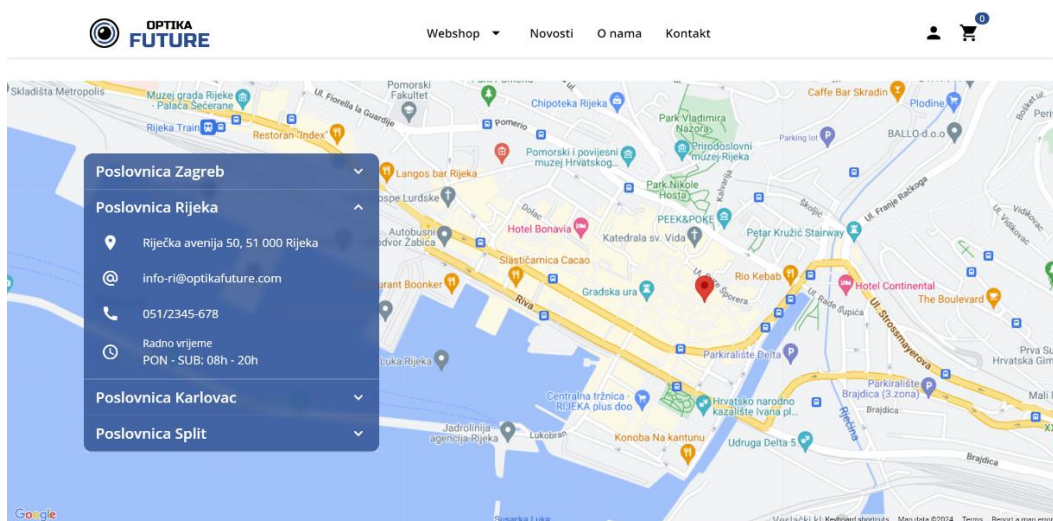
Nakon pregleda novosti na redu je dio s korisničkim recenzijama koji je prikazan na slici 21. Ovaj segment pruža autentičan uvid u iskustva stvarnih korisnika, što može značajno utjecati na povjerenje i odluke potencijalnih kupaca. Recenzije, koje su vizualno predstavljene u obliku kartica, odražavaju stvarne povratne informacije od strane kupaca, pružajući tako transparentan pregled usluga i proizvoda koje tvrtka nudi. Na vrhu se nalazi prikaz prosječne ocjene, ukupan broj recenzija te opcija „Napiši osvrt“ koja korisnika vodi na stranicu za pisanje recenzija.

Svaka recenzija sastoji se od imena osobe, ocjene u obliku zvjezdica, naslova, teksta recenzije te naznake prije koliko vremena je recenzija napisana. Ako osoba ima profilnu sliku onda će se također i ona prikazati uz ime osobe. Dodatno, ovisno o platformi na kojoj je recenzija ostavljena, prikazat će se i logo te platforme (na primjer Google, Facebook i slično).



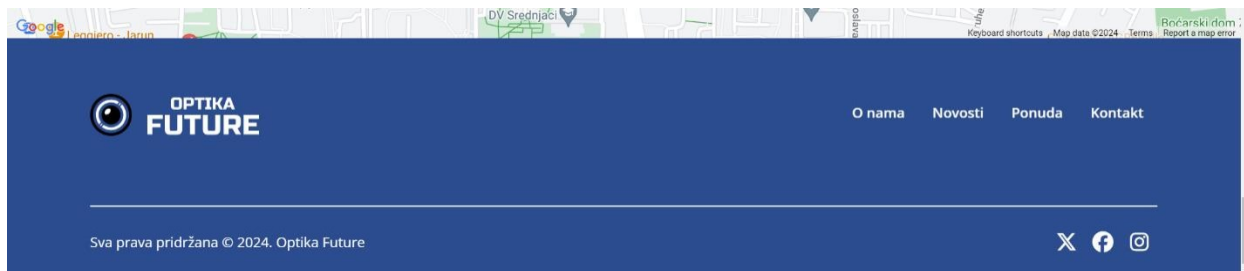
Slika 21 Prikaz sekcije korisničkih recenzija

Na kraju, prije samog podnožja stranice, nalazi se interaktivna karta koja služi kao digitalni vodič kroz fizičke poslovnice tvrtke (slika 22). Ovaj praktični alat omogućava korisnicima da brzo lociraju i dobiju osnovne informacije o svakoj poslovnici, uključujući adresu, kontakt informacije te radno vrijeme. Odabirom poslovnice prikazuju se njeni osnovni podaci kao i točna lokacija na Google karti.



Slika 22 Interaktivna Google karta s prikazom točnih lokacija i osnovnih informacija o poslovnicama

Podnožje (engl. footer) web aplikacije jednostavnog je dizajna, a sastoji se od loga tvrtke, navigacijskih elemenata, izjave o autorskim pravima te poveznica na društvene mreže. Slika 23 prikazuje podnožje web aplikacije „Optika Future“.



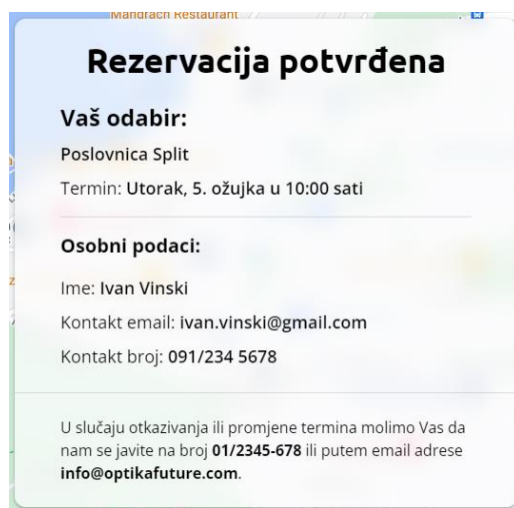
Slika 23 Podnožje web aplikacije "Optika Future"

5.1.2. Rezervacija besplatnog pregleda

Web aplikacija „Optika Future“ nudi posebnu stranicu posvećenu rezervaciji besplatnog pregleda vida prikazane na slici 25, kombinirajući funkcionalnost i informativni sadržaj kako bi korisnicima olakšala zakazivanje termina. Na vrhu stranice nalazi se jasno postavljena i privlačna pozivnica za rezervaciju, uz naglašavanje koristi koje klijenti imaju od ovog besplatnog pregleda, kao što su stručna dijagnostika, personalizirani savjeti, pristup najnovijoj tehnologiji i usluga bez obaveza.

Ispod ovog segmenta, posjetitelji mogu izabrati poslovnicu, datum i vrijeme za pregled pomoću intuitivnog korisničkog sučelja koje vodi korisnike kroz proces rezervacije korak po korak. Karta prikazuje lokaciju svake poslovnice, olakšavajući korisnicima da pronađu najbližu lokaciju i da se upoznaju s njezinim radnim vremenom i kontakt podacima.

Rezervacijom termina prikazuje se poruka potvrde rezervacije s detaljima o odabranom terminu te unesenim osobnim podacima za kontakt kako je prikazano na slici 24. Dodatno se ispod prikazuje upute za otkazivanje ili promjenu termina.



Slika 24 Prikaz poruke potvrde uspješne rezervacije termina za pregled vida

Rezervirajte svoj termin za besplatan pregled vida

Naši iskusni optometristi koriste najnoviju tehnologiju kako bi vam pružili temeljitu i preciznu dijagnostiku vida. Bez obzira na to trebate li naočale, kontaktne leće ili samo želite provjeriti zdravlje vaših očiju, naš tim stručnjaka je ovdje da vam pomogne.



Zašto rezervirati besplatan pregled?

Stručna dijagnostika

Iskoristite naše iskustvo i profesionalne uređaje za točnu procjenu vašeg vida.



Personalizirani savjeti

Dobit ćete personalizirane preporuke za vaš vizualni komfor i zdravije oči.



Najnovija tehnologija

Koristimo samo najmodernije metode i uređaje za pregled vašeg vida.



Bez obaveza

Naš besplatni pregled vida nudi vam uvid u vaše trenutno stanje vida bez ikakvih obaveza.



Rezervacija termina

- ✓ Odabrana je poslovnica: Split
- ✓ Odabir datuma
- 📅 **utorak, 5. ožujak**
- Natavi na odabir vremena** Natrag
- ⌚ Odabir vremena
- 👤 Unos osobnih podataka

Poslovnica Split
 Splitska avenija 21, 21 000 Split
 info-st@optikafuture.com
 021/2345-678
 PON - SUB: 08h - 20h

**OPTIKA
FUTURE**

O nama Novosti Ponuda Kontakt

Sva prava pridržana © 2024. Optika Future

✕ 🌐 📷

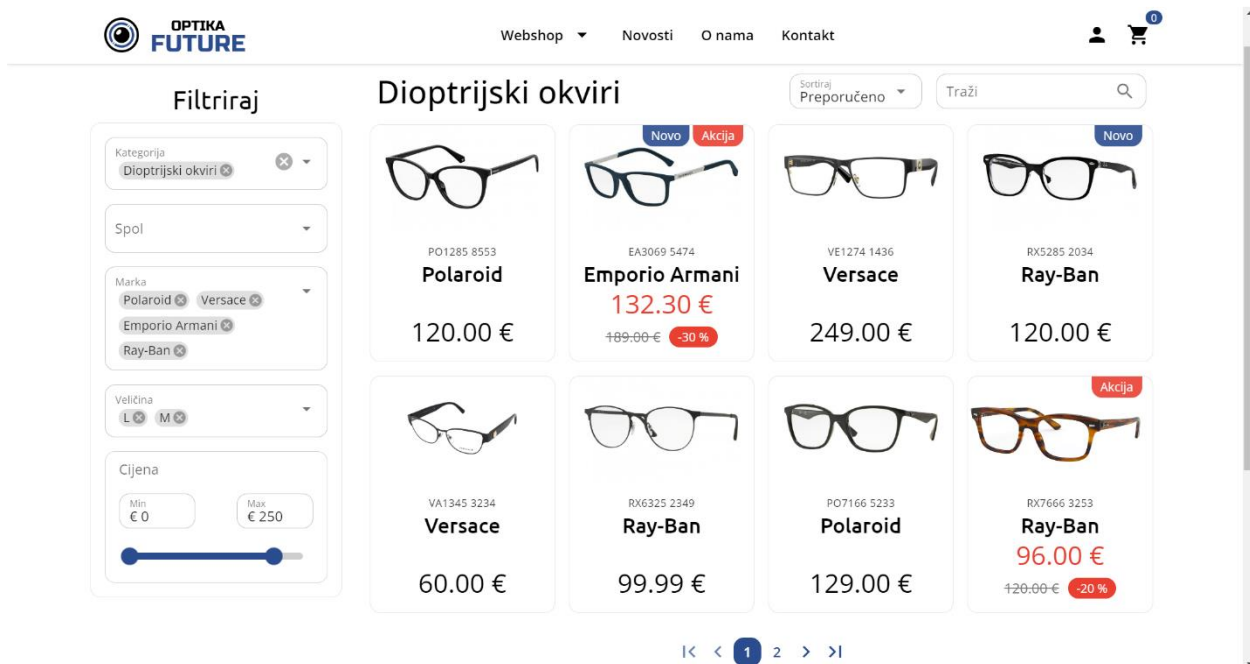
Slika 25 Stranica za rezervaciju termina besplatnog pregleda vida

5.1.3. Stranica ponude proizvoda

Stranica ponude proizvoda (slika 26) izrađena je tako da omogućava korisnicima jednostavno pregledavanje i filtriranje proizvoda prema različitim kriterijima. S lijeve strane, alat za filtriranje nudi opcije selekcije proizvoda prema kategoriji, spolu, marki, veličini, te cjenovnom rasponu, što korisnicima omogućava da prilagode pretragu svojim specifičnim potrebama i preferencijama. U gornjem desnom kutu postoji opcija sortiranja i pretrage proizvoda. Zadana vrijednost sortiranja je „Preporučeno“, a omogućeno je još i sortiranje prema cijena te abecedno.

Središnji dio stranice predstavlja proizvode kroz čiste i organizirane kartice koje sadrže ključne informacije kao što su naziv i model, fotografija proizvoda, cijena, te oznake za nove i proizvode na akciji. Promocije su jasno istaknute kroz cjenovne popuste, privlačeći pažnju na posebne ponude i potičući kupovinu.

Navigacija kroz stranicu s proizvodima je intuitivna, omogućujući korisnicima da lako prelaze između stranica ponude, što je označeno brojevima stranica na dnu. Takav pristup dizajnu stranice za ponudu proizvoda omogućava korisnicima da brzo i učinkovito pronađu proizvode koji ih zanimaju, unapređujući njihovo online iskustvo kupovine.



Slika 26 Stranica ponude proizvoda

Klikom na karticu proizvoda dolazimo do stranice pojedinačnog proizvoda koja je prikazana na slici 27. Ovaj prikaz nudi detaljne informacije o odabranom proizvodu, kao što su model, naziv, opis, cijena, karusel slika te procijenjeno vrijeme dostave. Ovisno o proizvodu nudi se i opcija odabira boje i veličine prije dodavanja u košaricu. Čak i neregistrirani korisnici mogu dodavati proizvode u košaricu, no prije kupnje potrebno je proći kroz proces registracije što će biti detaljnije opisano u poglavlju „5.1.5. Registracija i prijava“.

Dodatno se pri vrhu stranice nalazi i takozvana „breadcrumbs“ navigacija koja korisnika dodatno informira na kojem se mjestu web aplikacije nalazi. Ova dodatna navigacija je također i klikabilna kako bi se korisnik na jednostavan način mogao vratiti na željeno mjesto u aplikaciji.



PLDD372 807

Polaroid

Jeste li znali da istovremeno možete izgledati profesionalno i moderno? Ove bezvremenske dioptrijske naočale Polaroid PLDD372 807 to omogućuju. Njihov puni okrugli crni okvir možda nije pretjeran, no ipak privlači pažnju. Izrađen od lagane plastike, ovaj model izgleda nevjerovatno na licu četvrtastog i srcolikog oblika. Odmah nabavite savršen dodatak svojoj garderobi.

Boja



Veličina

**120.00 €** 1-3 radna dana[Dodaj u košaricu](#)

Slika 27 Stranica prikaza pojedinačnog proizvoda

5.1.4. Stranica novosti

Slika 28 prikazuje stranicu „Novosti“ na kojoj posjetitelji mogu pronaći najnovije članke i informacije koje se tiču svijeta optike i zdravlja očiju. Svaki članak prati relevantna vizualna ilustracija ili fotografija koja doprinosi zainteresiranosti i razumijevanju sadržaja. Posjetitelji mogu kliknuti na „Čitaj više“ kako bi se dublje uronilo u temu i saznalo više.

Novosti i zanimljivosti



18. 02. 2024.

Vodič kroz trendove naočala za 2024: Budite u toku s najnovijim stilovima

Kako se približavamo 2024., svijet naočala doživljava revoluciju ne samo u funkcionalnosti, već i u modnom izričaju. Od vintage okvira koji odišu nostalgijom do futurističkih dizajna koji izgledaju kao da su stigli ravno iz znanstveno-fantastičnog romana, izbor naočala nikada nije bio uzbudljiviji. U ovom blogu, istražiti ćemo najnovije trendove koji će obilježiti 2024. i kako možete pronaći par koji najbolje odgovara vašem stilu i potrebama.

Trendovi 2024

[Čitaj više >](#)

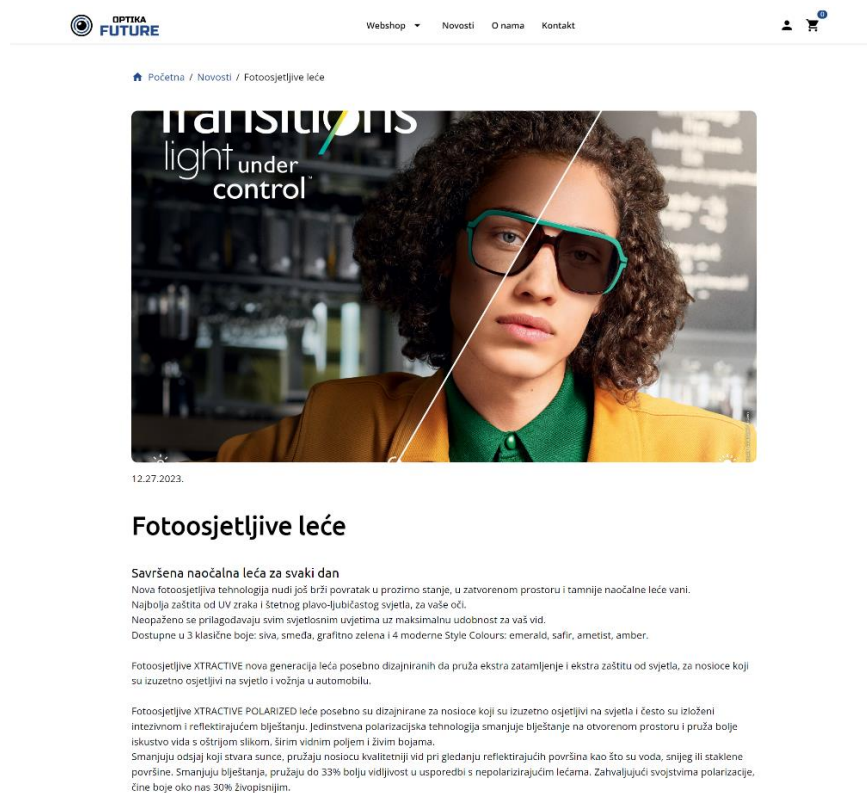
26. 01. 2024.

Fotoosjetljive leće

Jeste li ikada poželjeli da vaše naočale mogu čitati vaš um? Zamislite leće koje se automatski prilagođavaju svjetlini sunca, štiteći vaše oči i pružajući vam kristalno jasan vid bez potrebe za

Slika 28 Stranica "Novosti"

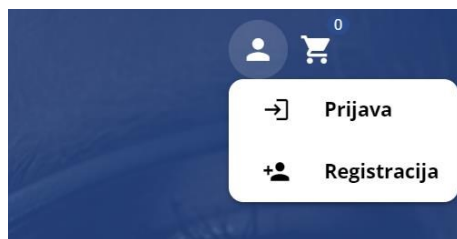
Odabirom opcije „Čitaj više“ otvara se nova stranica koja sadrži dodatne informacije o samoj temi što je vidljivo na slici 29. Kao i kod pregleda pojedinačnog proizvoda i ovdje se nalazi dodatna „breadcrumbs“ navigacija za lakše kretanje aplikacijom.



Slika 29 Stranica pojedinačne novosti, odnosno objave

5.1.5. Registracija i prijava

Da bi korisnik mogao kupovati proizvode prvo je potrebno obaviti registraciju. Odabirom ikone osobe iz desnog kuta navigacijske trake otvara se padajući izbornik s opcijama „Prijava“ i „Registracija“ (slika 30).



Slika 30 Padajući izbornik s opcijama Prijava i Registracija

Odabirom registracije dolazimo na stranicu za kreiranje računa, odnosno registraciju. Na slici 31 je prikazana forma za registraciju korisničkog računa na web stranici. Dizajn forme je čist i intuitivan, s jasno označenim poljima za unos osnovnih podataka potrebnih za stvaranje novog računa. Forma traži od korisnika da unese ime i prezime, email adresu, lozinku te potvrdu lozinke, osiguravajući tako da su sve

informacije točne i sigurne. Gumb „Registriraj se“ istaknut je plavom bojom koja se slaže s bojom brenda, što korisnicima olakšava identifikaciju sljedećeg koraka u procesu registracije.

Dodatno, nudi se opcija „Nastavi uz Google“, omogućavajući korisnicima brzu registraciju koristeći već postojeće Google račune, što proces čini bržim i efikasnijim za one koji preferiraju ovu metodu.

Na dnu forme nalazi se i veza za korisnike koji već imaju račun, što pruža brz pristup formi za prijavu za postojeće korisnike. Ova funkcionalnost je korisna jer spaja procese registracije i prijave, čineći korisničko iskustvo glatkim i neometanim.

Slika 31 Forma za registraciju

Na slici 32 prikazana je forma za prijavu (autentifikaciju) korisnika. Forma je jasno strukturirana kako bi korisnicima pružila brz i jednostavan način prijave na svoj račun. Polja za unos su standardna i zahtijevaju email adresu i lozinku, što je uobičajeni postupak za pristupanje osobnom računu.

Opcija „Zapamti me“ je ponuđena kako bi se korisnicima omogućilo da ostanu prijavljeni na svojim uređajima, čime se izbjegava potreba za ponovnim unosom podataka pri svakoj posjeti web stranici. Za one koji su možda zaboravili svoju lozinku, veza „Zaboravili ste lozinku?“ nudi brz način oporavka pristupa računu.

Kao i kod procesa registracije i ovdje postoji opcija „Nastavi uz Google“ za prijavu pomoću Google računa te opcija „Registriraj se“ za nove korisnike koji još nemaju kreiran račun.


Prijava

Email

Lozinka

Zapamti me [Zaboravili ste lozinku?](#)

Prijavi se

 Nastavi uz Google

Nemaš račun? [Registriraj se](#)

Slika 32 Forma za prijavu korisnika

5.2. Klijenti (registrirani korisnici)

Klijente smatramo one korisnike koji su prošli proces registracije. Klijenti imaju sve mogućnosti kao i gosti, no s dodatnim privilegijama kao što je kupnja proizvoda, pregled povijesti narudžba i slično, što će biti opisano u nastavku.

5.2.1. Stranica „Moj račun“





Uspješnom prijavom korisniku će biti prikazana stranica „Moj račun“ (slika 33). Ovdje je korisnicima omogućen pristup i upravljanje njihovim osobnim podacima. Ova korisnička nadzorna ploča je jasno organizirana i podijeljena u nekoliko segmenata za jednostavnu navigaciju: „Moji podaci“, „Moje adrese“, „Moje narudžbe“ i „Postavke“.

U dijelu „Moji podaci“, korisnik može vidjeti i uređivati svoje osnovne informacije kao što su ime, prezime, broj mobitela i email adresa. Ovo je korisno za održavanje ažurnih podataka za komunikaciju i obradu narudžbi. Svi uneseni podaci mogu se spremi korištenjem plavog gumba „Spremi“, što osigurava da su sve promjene uspješno zabilježene.

Stranica je dizajnirana s fokusom na korisniku prijateljski pristup, omogućujući jednostavne izmjene i pregled informacija bez nepotrebnog kompliciranja procesa. Pristup ovoj stranici omogućuje korisnicima da imaju veću kontrolu nad svojim računom, pridonoseći pozitivnom korisničkom iskustvu.

Putem stranice „Moje narudžbe“ korisnik ima mogućnost pregledati svoje prošle narudžbe te njihov status (u obradi, poslano, dostavljeno). Također postoji i opcija „Postavke“ putem koje korisnik može obrisati svoj račun ili zatražiti promjenu lozinke.

Moj račun

-  Moji podaci
-  Moje adrese
-  Moje narudžbe
-  Postavke

Moji podaci

Ime

Ivan

Prezime

Vinski

Broj mobitela

091 234 5678

Email adresa





ivanvinski@gmail.com

Spremi

Slika 33 Stranica "Moj račun"

Stranica „Moje adrese“ (slika 34) unutar korisničkog računa omogućava korisnicima da upravljaju i spreme svoje adrese za dostavu i naplatu. Ovo je korisno za brže i učinkovitije procesuiranje budućih narudžbi, jer korisnik ne mora ponovno unositi svoju adresu svaki put kada želi obaviti kupnju novih proizvoda.

Moj račun

-  Moji podaci
-  Moje adrese
-  Moje narudžbe
-  Postavke

Moje adrese

+ Nova adresa

Adresa 1

Ulica i kućni broj

Moja adresa 5

Grad

Rijeka

Poštanski broj

51000

Država

Hrvatska

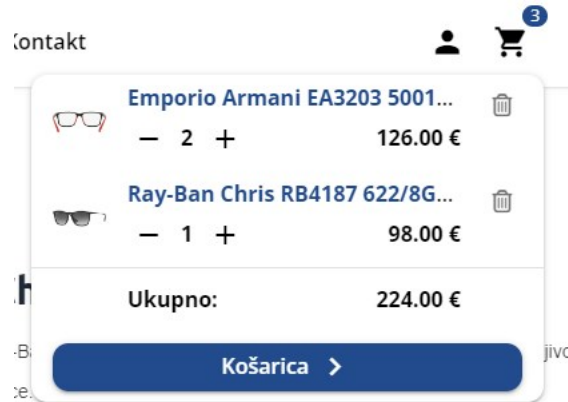
Uredi

 Obriši

Slika 34. Stranica „Moje adrese“

5.2.2. Stranica „Košarica“

Odabirom ikone košarice u gornjem desnom kutu navigacije korisniku se prikaže pregled proizvoda dodanih u istu, kako je prikazano na slici 35. Opcijama plus i minus moguće je povećati ili smanjiti broj određenog proizvoda u košarici, dok se odabirom ikone kante za smeće proizvod uklanja iz košarice. Iznad ikone košarice dodatno se nalazi i indikator koji označava ukupan broj proizvoda dodanih u košaricu.



Slika 35 Pregled košarice u padajućem izborniku

Odabirom opcije „Košarica“ otvara se zasebna stranica košarice koja sadrži iste funkcionalnosti kao i prikaz u padajućem izborniku. Zasebna stranica košarice prikazana je na slici 36. Odabirom opcije „Nastavi“ dolazimo na stranicu za odabir adrese nakon koje je moguće platiti te naručiti proizvode koji se nalaze unutar košarice.

Košarica



Slika 36 Pregled košarice na zasebnoj stranici

5.3. Administratori

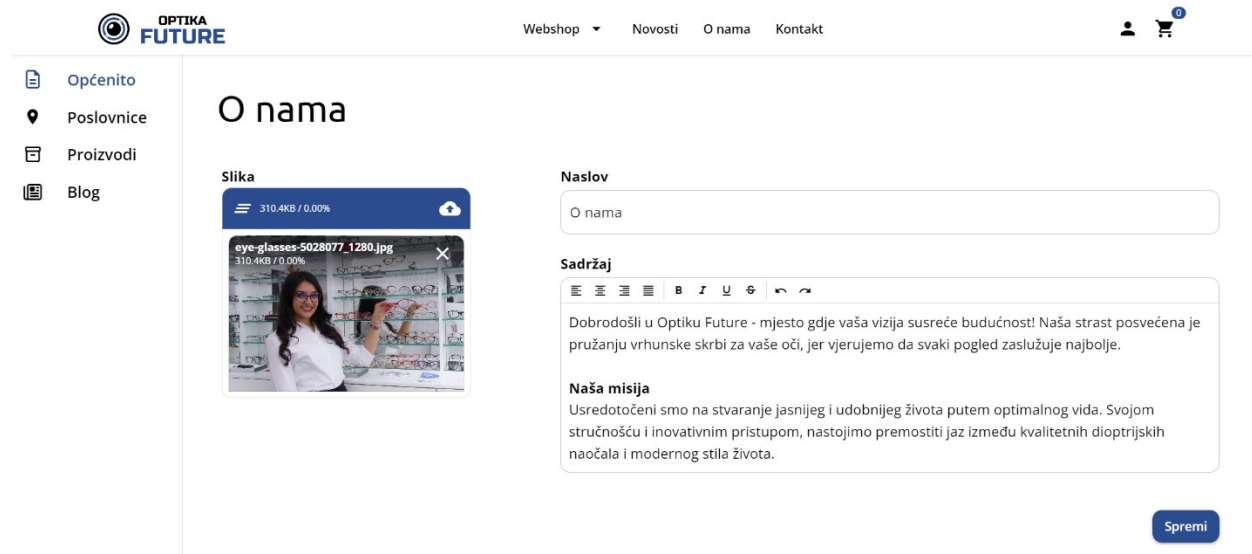
5.3.1. Stranica „Općenito“

Slika 37 prikazuje sučelje za administrativno uređivanje sekcije „O nama“ koje se nalazi na početnoj stranici. Kroz ovaj sustav za upravljanje sadržajem (engl. Content Management System, CMS), administrator web stranice može ažurirati i uređivati sadržaj koji se prikazuje posjetiteljima.

Na lijevoj strani sučelja nalaze se opcije za navigaciju kroz različite administrativne sekcije kao što su „Općenito“, „Poslovnice“, „Proizvodi“ i „Blog“. To administratorima omogućava brz pristup različitim dijelovima web stranice za ažuriranje sadržaja.

Centralni dio ekrana je posvećen uređivanju sekcije „O nama“. Administrator ima mogućnost da unese naslov i sadržaj stranice. Ispod polja za unos sadržaja nalazi se gumb „Spremi“ koji omogućuje administratoru da sačuva napravljene izmjene. To je završni korak u procesu uređivanja, nakon kojeg će ažurirani sadržaj biti vidljiv posjetiteljima web aplikacije. Na slici 37 je također vidljiva minijatura slike koja će se pojaviti u sekciji „O nama“, pružajući vizualni element koji upotpunjuje tu sekciju.

Ovo administratorsko sučelje je bitno za održavanje aktualnosti i relevantnosti web stranice, omogućujući brzo i efikasno upravljanje sadržajem.



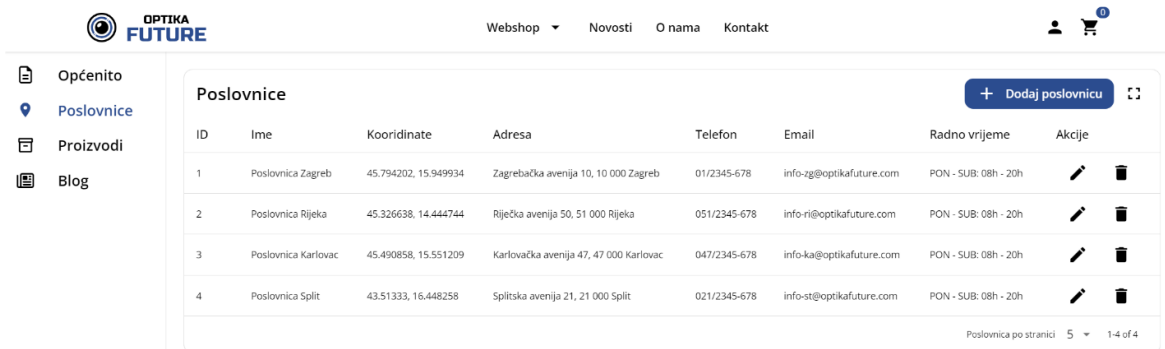
Slika 37 Administrativno sučelje za uređivanje sekcije "O nama"

5.3.2. Stranica „Poslovnice“

Na slici 38 je prikazan administratorski prikaz sučelja za upravljanje poslovnica. Administratorska ploča omogućuje pregled i uređivanje detaljnih informacija o svakoj poslovnici, uključujući ime poslovnice, geografske koordinate, adresu, kontakt telefon, e-mail adresu i radno vrijeme.

Administrator ima sposobnost da dodaje nove poslovnice koristeći opciju „Dodaj poslovnici“, kao i da uređuje ili briše postojeće unose korištenjem ikona olovke (za uređivanje) i kante za smeće (za brisanje) u kolumni „Akcije“. Također je omogućena i opcija prikaza tablice preko cijelog zaslona (engl. fullscreen) odabirom ikone u gornjem desnom kutu tablice, čime se postiže bolja preglednost samih podataka o poslovnicama. Ovo pruža fleksibilnost i lakoću održavanja točnosti informacija koje su važne za korisnike koji žele posjetiti fizičke lokacije. Na slici 39 prikazan je obrazac za uređivanje postojeće ili dodavanje nove poslovnice.

Ovakav način upravljanja informacijama o poslovnicama je neophodan kako bi korisnici uvijek imali pristup najnovijim i točnim podacima, a također i kako bi se osiguralo da web aplikacija pruža pouzdane i ažurirane informacije.

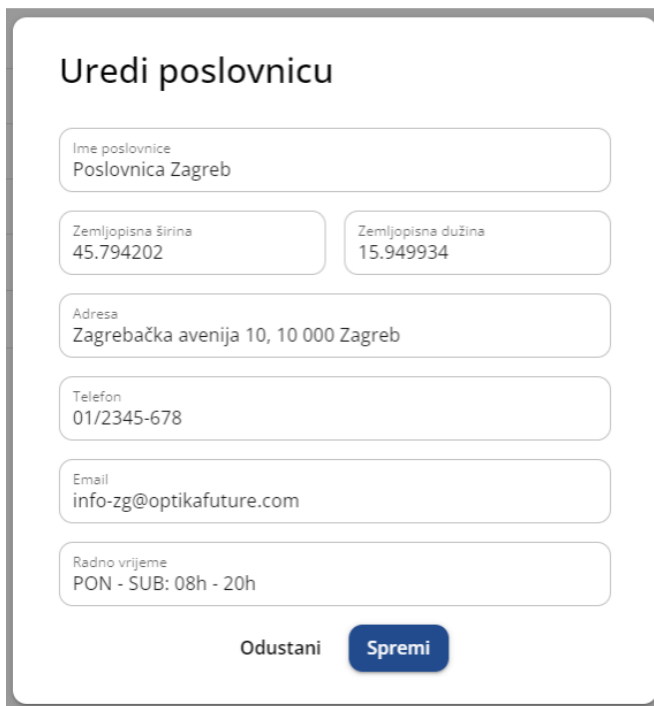


The screenshot shows the administrator interface for 'OPTIKA FUTURE'. At the top, there is a navigation menu with 'Webshop', 'Novosti', 'O nama', and 'Kontakt'. On the right, there are icons for a user profile and a shopping cart. A left sidebar contains navigation options: 'Općenito', 'Poslovnice', 'Proizvodi', and 'Blog'. The main content area is titled 'Poslovnice' and features a '+ Dodaj poslovnici' button and a fullscreen icon. Below this is a table with the following data:

ID	Ime	Koordinate	Adresa	Telefon	Email	Radno vrijeme	Akcije
1	Poslovnica Zagreb	45.794202, 15.949934	Zagrebačka avenija 10, 10 000 Zagreb	01/2345-678	info-zg@optikafuture.com	PON - SUB: 08h - 20h	
2	Poslovnica Rijeka	45.326638, 14.444744	Riječka avenija 50, 51 000 Rijeka	051/2345-678	info-ri@optikafuture.com	PON - SUB: 08h - 20h	
3	Poslovnica Karlovac	45.490858, 15.551209	Karlovačka avenija 47, 47 000 Karlovac	047/2345-678	info-ka@optikafuture.com	PON - SUB: 08h - 20h	
4	Poslovnica Split	43.51333, 16.448258	Splitaska avenija 21, 21 000 Split	021/2345-678	info-st@optikafuture.com	PON - SUB: 08h - 20h	

At the bottom right of the table, there is a pagination indicator: 'Poslovnica po stranici 5 1-4 of 4'.

Slika 38 Administratorsko sučelje za upravljanje poslovnicama



The screenshot shows the 'Uredi poslovnici' (Edit branch) form. It contains the following fields and values:

- Ime poslovnice: Poslovnica Zagreb
- Zemljopisna širina: 45.794202
- Zemljopisna dužina: 15.949934
- Adresa: Zagrebačka avenija 10, 10 000 Zagreb
- Telefon: 01/2345-678
- Email: info-zg@optikafuture.com
- Radno vrijeme: PON - SUB: 08h - 20h

At the bottom of the form, there are two buttons: 'Odustani' (Cancel) and 'Spremi' (Save).

Slika 39 Obrazac za uređivanje/dodavanje poslovnice

5.3.3. Stranica „Proizvodi“

Na slici 40 prikazano je administratorsko sučelje za upravljanje proizvodima gdje administrator ima mogućnost pregleda i uređivanja popisa proizvoda. Sučelje prikazuje tablicu s popisom proizvoda, gdje svaki proizvod ima više atributa koji su jasno prikazani u stupcima: naziv, cijena, popust u postotcima, konačna cijena, slika, količina, kategorija, spol, tagovi te akcije koje se mogu poduzeti za svaki proizvod.

Stupac „Akcije“ sadrži ikone za uređivanje i brisanje pojedinog proizvoda, što administratorima omogućuje lako ažuriranje ili uklanjanje proizvoda iz ponude. U skroz lijevom stupcu tablice nalaze se potvrdni okviri (engl. checkbox) koji omogućuju administratorima označavanje više proizvoda odjednom kako bi lako mogli izvršiti izmjene ili obrisati veći broj proizvoda istovremeno. Gumb „Dodaj novi proizvod“ služi za unos novih proizvoda u katalog na web aplikaciji. Dodatno, postoji opcija prikaza tablice preko cijelog zaslona (engl. full screen) što povećava preglednost tablice omogućavajući administratorima jednostavnije upravljanje proizvodima.

Funkcionalnosti kao što su filtriranje stupaca, traženje specifičnih proizvoda kroz pretraživač, te opcije za navigaciju kroz stranice proizvoda, pružaju administratoru učinkovite alate za upravljanje asortimanom proizvoda. Ovo sučelje je ključno za održavanje ažurne i atraktivne ponude za posjetitelje web aplikacije.

<input type="checkbox"/>	Naziv	Cijena (€)	Popust (%)	Konačna cijena (€)	Slika	Količina	Kategorija	Spol	Tagovi	Akcije
<input type="checkbox"/>	Emporio Armani ...	70.00	10	63.00		6	Dioptrijski okviri	Dječje	Promocija	
<input type="checkbox"/>	Ray-Ban Chris RB...	98.00	0	98.00		0	Sunčane naočale	Muške		
<input type="checkbox"/>	Polaroid PLDD37...	50.00	0	50.00		20	Dioptrijski okviri	Ženske	U trendu	
<input type="checkbox"/>	Ray-Ban Clubmas...	116.00	20	92.80		15	Dioptrijski okviri	Unisex	Akcija	
<input type="checkbox"/>	Polaroid PLD6012...	60.00	0	60.00		14	Sunčane naočale	Unisex	Novo	

Slika 40 Administratorsko sučelje za upravljanje proizvodima

6. Zaključak

U ovom diplomskom radu razvijena je web aplikacija za oglašavanje i trgovinu optikom, koristeći suvremene tehnologije poput Nuxta, Quasara, Tailwind CSSa i Firebasea.

Projekt je uspješno demonstrirao kako se kroz primjenu odabranih tehnologija i alata može postići responzivna, sigurna i korisnički privlačna web platforma. Implementacija funkcionalnosti kao što su pregled proizvoda, upravljanje košaricom, proces rezervacije termina, uređivanje i dodavanje proizvoda, poslovnica i objava učinjena je s naglaskom na intuitivnosti i pristupačnosti, što je ključno za zadovoljstvo krajnjih korisnika.

Osim toga, ovaj rad je otvorio raspravu o važnosti detaljne analize i razumijevanja korisničkih potreba, kao i o prilagodljivosti u dizajnu i razvoju web rješenja. Kroz kritičku evaluaciju korištenih tehnologija i metoda, rad pruža vrijedne uvide koji mogu biti korisni za buduće projekte u sličnim područjima.

Kroz projekt su istražene i implementirane najbolje prakse u dizajnu i razvoju web aplikacija, što je rezultiralo u aplikaciji koja ne samo da zadovoljava, već i premašuje očekivanja korisnika. Ovaj rad služi kao primjer kako tehnološka rješenja mogu unaprijediti poslovanje u optičkoj industriji, pružajući korisnicima napredne funkcionalnosti i poboljšavajući ukupno iskustvo kupovine.

Zaključno, ovaj diplomski rad ne samo da potvrđuje važnost tehnološke inovacije u unapređenju poslovnih procesa i korisničkih iskustava, već i ističe kako kontinuirano učenje i prilagodba novim tehnološkim trendovima ostaje neophodno za ostvarivanje konkurentske prednosti u dinamičnom poslovnom okruženju. Nadamo se da će ovaj rad poslužiti kao koristan izvor informacija i inspiracije za buduće projekte u dinamičnom i stalno evoluirajućem polju web razvoja.

Analiza i iskustva dobivena tijekom ovog projekta mogu poslužiti kao temelj za daljnje istraživanje, posebno u pogledu optimizacije korisničkog iskustva i integracije naprednih tehnologija poput umjetne inteligencije i virtualne stvarnosti u kontekstu online trgovine.

7. Popis literature i izvora

1. Ccari, Paul. "Linting in TypeScript Using ESLint and Prettier." LogRocket Blog. 26. rujna 2023. Izvor prema stanju od 3. veljače 2024. <https://blog.logrocket.com/linting-typescript-eslint-prettier/#avoiding-conflicts-working-eslint-prettier>.
2. Chu, Johnson. 2023. "Volar: A New Beginning." The Vue Point. 8. veljače 2023. Izvor prema stanju od 21. veljače 2024. <https://blog.vuejs.org/posts/volar-a-new-beginning>.
3. Farrag, Adham. "Nuxtr VSCode." GitHub. 24. siječnja 2024. Izvor prema stanju od 21. veljače 2024. <https://github.com/nuxtrdev/nuxtr-vscode#nuxtr-vscode>.
4. Fitzgerald, Anna. "What Is a CDN? Understanding Content Delivery Networks." HubSpot, 5. siječnja 2024. Izvor prema stanju od 19. siječnja 2024. <https://blog.hubspot.com/website/what-is-a-cdn-definition>.
5. Gupta, Shivam. "ESLINT: What, Why, When, How." DEV Community. 19. lipnja 2021. Izvor prema stanju od 19. siječnja 2024. <https://dev.to/shivambmgupta/eslint-what-why-when-how-5f1d>.
6. "IntelliSense in Visual Studio Code." Visual Studio Code. 2. siječnja 2024. Izvor prema stanju od 21. veljače 2024. <https://code.visualstudio.com/docs/editor/intellisense>.
7. Jagodić, Barbara Slade. "Što Je Call-to-Action? Fraza Koja Povećava Prihode!" Slade.hr. 19. srpnja 2022. Izvor prema stanju od 6. veljače 2024. <https://www.slade.hr/blog/call-to-action/>.
8. Long, James. "A Prettier JavaScript Formatter." 10. siječnja 2017. Izvor prema stanju od 21. veljače 2024. <https://archive.jlongster.com/A-Prettier-Formatter>.
9. Modi, Anjali. "Quasar Framework Demystified: A Beginner's Journey." Bytes Technolab. 26. rujna 2023. Izvor prema stanju 18. siječnja 2024. <https://www.bytestechnolab.com/blog/uncover-quasar-framework-simplified-beginners-guide/>.
10. Nandaniya, Hamir. "A Guide to Component-Based Architecture: Features, Benefits and More." Maruti Techlabs. 20. studenog 2023. Izvor prema stanju od 21. veljače 2024. <https://marutitech.com/guide-to-component-based-architecture/>.
11. "Nuxt Overview." Sanity.io. 18. siječnja 2024. Izvor prema stanju od 19. siječnja 2024. <https://www.sanity.io/glossary/nuxt>.
12. Olawanle, Joel. "What Is TypeScript? A Comprehensive Guide." Kinsta®. 10. listopada 2023. Izvor prema stanju od 21. siječnja 2024. <https://kinsta.com/knowledgebase/what-is-typescript/>.
13. Özşahan, Hatice. "What Is Firebase? Learn The Basics From A to Z." Resmo. 18. svibnja 2023. Izvor prema stanju od 17. siječnja 2024. <https://www.resmo.com/blog/what-is-firebase>.
14. Rouse, Margaret. "Parser." Techopedia. 20. prosinca 2022. Izvor prema stanju od 2. veljače 2024. <https://www.techopedia.com/definition/3854/parser>.
15. "Swiper - the Most Modern Mobile Touch Slider." n.d. Swiper. Izvor prema stanju od 19. siječnja 2024. <https://swiperjs.com/>.
16. Sacolick, Isaac. "What Is CI/CD? Continuous Integration and Continuous Delivery Explained." InfoWorld. 4. prosinca 2023. Izvor prema stanju od 20. siječnja 2024. <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>.
17. Tailwind Labs. "Tailwindcss Intellisense." GitHub. 29. siječnja 2024. Izvor prema stanju od 21. veljače 2024. <https://github.com/tailwindlabs/tailwindcss-intellisense>.
18. Technologies, ultroNeous. The Importance of Agile Methodology in Software Development. LinkedIn. 12. svibnja 2023. Izvor prema stanju od 14. siječnja 2024.

<https://www.linkedin.com/pulse/importance-agile-methodology-software-development-ultroneous>.

19. Zohair Mustafeez, Anusheh. "What Is Visual Studio Code?" Educative. Izvor prema stanju 18. siječnja 2024. <https://www.educative.io/answers/what-is-visual-studio-code>.

8. Popis priloga

1. Poveznica na GitHub repozitorij s izvornim kodom aplikacije: <https://github.com/IvanVinski/optika-future-diplomski>
2. Poveznica na web aplikaciju „Optika Future“: <https://optika-future.web.app/>

9. Popis slika

Slika 1 Ilustracija agilnog razvojnog procesa (Technologies 2023)	7
Slika 2 Početna stranica novog Nuxt projekta	19
Slika 3 Konfiguracija quasar modula unutar nuxt.config.ts datoteke	20
Slika 4 quasar.variables.scss datoteka	20
Slika 5 Konfiguracijska datoteka tailwind.config.js	21
Slika 6 Primjer korištenja tailwind "utility" klasa u implementaciji uvodne sekcije početne stranice	21
Slika 7 Primjer korištenja Swiper biblioteke	23
Slika 8 Primjer korištenja vue3-google-map biblioteke	24
Slika 9 Primjer dodavanja Trustmary widget-a u Nuxt aplikaciju	25
Slika 10 Pretier konfiguracijska datoteka .prettierrc	26
Slika 11 Konfiguracija ESLinta u .eslintrc.js datoteci	27
Slika 12 Konfiguracija vuefire paketa unutar nuxt.config.ts datoteke	28
Slika 13 Popis odabranih opcija prijave u Firebase konzoli	29
Slika 14 Početna stranica web aplikacije "Optika Future"	31
Slika 15 Karusel izdvojene ponude na početnoj stranici	32
Slika 16 Karusel izdvojene ponude na mobilnim uređajima	33
Slika 17 Sekcija "O nama" na početnoj stranici	33
Slika 18 Sekcija "Kontakt" na početnoj stranici	34
Slika 19 Prikaz poruke zahvalnosti nakon slanja upita	34
Slika 20 Sekcija "Novosti" na početnoj stranici	35
Slika 21 Prikaz sekcije korisničkih recenzija	36
Slika 22 Interaktivna Google karta s prikazom točnih lokacija i osnovnih informacija o poslovnica	36
Slika 23 Podnožje web aplikacije "Optika Future"	37
Slika 24 Prikaz poruke potvrde uspješne rezervacije termina za pregled vida	37
Slika 25 Stranica za rezervaciju termina besplatnog pregleda vida	38
Slika 26 Stranica ponude proizvoda	39
Slika 27 Stranica prikaza pojedinačnog proizvoda	40
Slika 28 Stranica "Novosti"	40

Slika 29 Stranica pojedinačne novosti, odnosno objave	41
Slika 30 Padajući izbornik s opcijama Prijava i Registracija	41
Slika 31 Forma za registraciju	42
Slika 32 Forma za prijavu korisnika	43
Slika 33 Stranica "Moj račun"	44
Slika 34. Stranica „Moje adrese“	44
Slika 35 Pregled košarice u padajućem izborniku	45
Slika 36 Pregled košarice na zasebnoj stranici	45
Slika 37 Administrativno sučelje za uređivanje sekcije "O nama"	46
Slika 38 Administratorsko sučelje za upravljanje poslovnica	47
Slika 39 Obrazac za uređivanje/dodavanje poslovnice	47
Slika 40 Administratorsko sučelje za upravljanje proizvodima.....	48

10. Popis tablica

Tablica 1 Usporedba ESLint-a i Pretier-a (Ccarl 2023)	26
---	----