

# Web aplikacija za prodaju nekretnina

---

**Koturić, Ivan**

**Undergraduate thesis / Završni rad**

**2024**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:658276>

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

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

*Repository / Repozitorij:*

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



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

**Sveučilišni studij Računarstvo**

**WEB APLIKACIJA ZA PRODAJU NEKRETNINA**

**Završni rad**

**Ivan Koturić**

**Osijek, 2024.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Ivan Koturić
<b>Studij, smjer:</b>	Sveučilišni prijediplomski studij Računarstvo
<b>Mat. br. pristupnika, god.</b>	R-4369, 11.10.2021.
<b>JMBAG:</b>	0165081854
<b>Mentor:</b>	izv. prof. dr. sc. Ivica Lukić
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Web aplikacija za prodaju nekretnina
<b>Znanstvena grana završnog rada:</b>	Informacijski sustavi (zn. polje računarstvo)
<b>Zadatak završnog rada:</b>	Razviti web aplikaciju za prodaju nekretnina, pretraživanje već objavljenih nekretnina. Registrirani korisnici mogu kupiti ili rezervirati nekretninu. Aplikacija će biti razvijena upotrebom React i NodeJS. Tema rezervirana za: Ivan Koturić
<b>Datum prijedloga ocjene završnog rada od strane mentora:</b>	17.09.2024.
<b>Prijedlog ocjene završnog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum potvrde ocjene završnog rada od strane Odbora:</b>	25.09.2024.
<b>Ocjena završnog rada nakon obrane:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:</b>	26.09.2024.

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

Osijek, 26.09.2024.

**Ime i prezime Pristupnika:**

Ivan Koturić

**Studij:**

Sveučilišni prijediplomski studij Računarstvo

**Mat. br. Pristupnika, godina upisa:**

R-4369, 11.10.2021.

**Turnitin podudaranje [%]:**

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za prodaju nekretnina**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ivica Lukić

i sumentora

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

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

Potpis pristupnika:

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
1.1. Zadatak rada .....	1
<b>2. PREGLED PODRUČJA TEME</b> .....	<b>2</b>
2.1. Zillow .....	2
2.2. Realtor.com .....	3
2.3. Booking.com .....	4
2.4. Njuškalo .....	5
2.5. CentarNekretnina .....	6
<b>3. ZAHTJEVI ZA APLIKACIJU</b> .....	<b>8</b>
3.1. HTML .....	9
3.2. CSS .....	10
3.3. MERN .....	12
3.4. MongoDB .....	13
3.5. Express.js .....	14
3.6. React .....	14
3.7. Node.js .....	15
<b>4. POSTUPAK IZRADE APLIKACIJE</b> .....	<b>16</b>
4.1. Baza podataka .....	16
4.2. Komponente .....	18
<b>5. PRIKAZ RADA APLIKACIJE I KORISNIČKOG SUČELJA</b> .....	<b>22</b>
5.1. Naslovnica .....	22
5.2. Nekretnina .....	22
5.3. Nekretnine .....	24
5.4. Omiljene nekretnine i Rezervacije .....	24
5.5. Dodavanje nekretnina .....	25
5.6. Autentifikacija i registracija korisnika .....	26

<b>6. ZAKLJUČAK.....</b>	<b>27</b>
<b>LITERATURA .....</b>	<b>28</b>
<b>SAŽETAK.....</b>	<b>29</b>
<b>ABSTRACT .....</b>	<b>30</b>
<b>PRILOZI.....</b>	<b>31</b>

# 1. UVOD

U današnje vrijeme, digitalizacija prodaje nekretnina postala je neizbježna zbog brzog razvoja tehnologije i širenja interneta. Tradicionalni načini kupnje i prodaje nekretnina, kao što su oglašavanje u novinama i korištenje fizičkih agencija, u suvremenom brzom načinu života postaju jako nepraktični i neučinkoviti. Ovi pristupi često zahtijevaju puno vremena, resursa i fizičke prisutnosti, što ih čini neprikladnima za suvremene kupce i prodavatelje koji žele brža i jednostavnija rješenja. Kupci i prodavatelji nekretnina sve više traže brža i pristupačnija rješenja za pretraživanje, pregledavanje i transakcije nekretnina.

Motivacija za rad na ovoj temi proizlazi iz prepoznatljivih prednosti koje digitalne platforme donose u svakodnevne transakcije, uključujući tržište nekretnina. Potrošači danas očekuju trenutačan pristup informacijama i mogućnost obavljanja poslova putem interneta, čime se eliminiraju prostorna i vremenska ograničenja. Kupci žele imati pristup velikom broju nekretnina s različitih tržišta, dok prodavatelji traže načine za dosezanje šire publike na učinkovit način.

U drugom poglavlju opisane su već postojeće platforme za kupoprodaju nekretnina ili rezervaciju smještaja u nekretninama te usporedba s temom završnog rada. Treće poglavlje sadrži zahtjeve za aplikaciju te kratak pregled tehnologija korištenih za izradu ove web aplikacije, a četvrto poglavlje opisuje konkretnu izradu web aplikacije. Peto poglavlje opisuje korisničko iskustvo prilikom korištenja izadene aplikacije.

Digitalne platforme omogućuju korisnicima brz i jednostavan pristup velikom broju oglasa, usporedbu ponuda i donošenje informiranih odluka. Uz to, digitalizacija smanjuje troškove i povećava učinkovitost procesa kupoprodaje. U ovom završnom radu rješava se problem modernizacije i digitalizacije tržišta nekretnina kroz razvoj web aplikacije za prodaju nekretnina.

## 1.1. Zadatak rada

Zadatak ovog završnog rada je izrada web aplikacije koja na jednostavan način omogućuje korisnicima pronalazak nekretnine koja odgovara njihovim potrebama. Web aplikacija omogućuje korisnicima pristup informacijama o nekretninama bilo kada i bilo gdje, pruža detaljne opise i vizualne prikaze nekretnina, te omogućuje interaktivno pretraživanje i filtriranje prema specifičnim kriterijima te opcije za zakazivanje pregleda nekretnine s prodavačima.

## 2. PREGLED PODRUČJA TEME

U današnjem svijetu postoji velik broj raznih web aplikacija koje nude usluge prodaje ili kupnje nekretnina. Zbog toga teško je stvoriti web aplikaciju koja se ističe ili ima neku posebnu funkcionalnost koja već nije drugdje implementirana. Svaka zemlja ima neku vrstu platforme koja nudi usluge kupnje ili prodaje nekretnina isključivo za tu zemlju, postoji mnogo platformi koje nude zakazivanje pregleda nekretnine itd. Neke od platformi koje će biti obrađene u ovome poglavlju su:

- Zillow
- Realtor.com
- Booking
- Njuškalo
- CentarNekretnina

### 2.1. Zillow

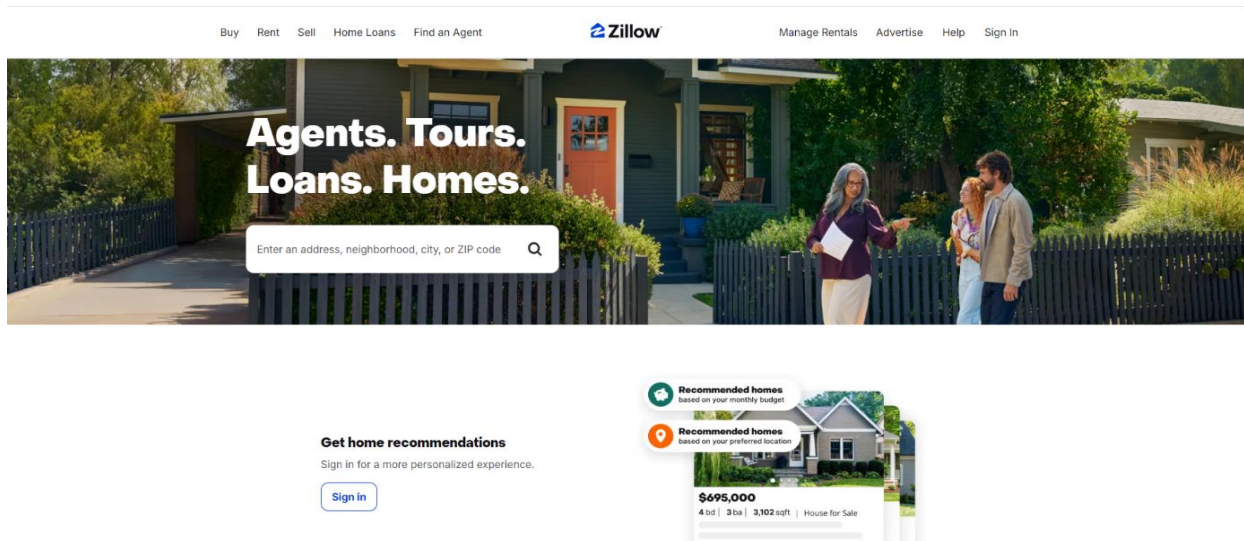
Zillow je jedna od najpopularnijih i najpoznatijih web platformi za nekretnine u Sjedinjenim Američkim Državama. Pokrenuta je 2006. godine te je promijenila način na koji ljudi pretražuju, kupuju, prodaju i iznajmljuju nekretnine. Zillow omogućuje korisnicima pregledanje velikog broja oglasa za nekretnine, usporedbu cijena, procjenu vrijednosti te velik broj informacija o samim nekretninama. Ključna značajka koja ju odvaja od drugih platformi nalik njoj je uvođenje *Zestimate* opcije, odnosno izračun ili procjena vrijednosti nekretnine korištenjem algoritma koji uzima u obzir podatke o prodajama, lokaciji, veličini i druge čimbenike. Iako se ne može smatrati preciznom, daje korisnicima barem približnu ideju o vrijednosti nekretnine [1].

Funkcionalnosti i značajke: Zillow nudi bogat skup funkcionalnosti koje olakšavaju proces kupnje i prodaje nekretnina. Platforma također omogućuje korisnicima filtriranje pretrage prema različitim kriterijima poput cijene, veličine, broja soba i lokacije. Interaktivne karte pomažu korisnicima vizualno pregledavati područje u kojem traže nekretninu, prikazujući dodatne informacije poput škola, trgovina i prometne povezanosti.

Zillow, kao i izrađena aplikacija, korisniku nudi slično korisničko sučelje s prikazom informacija o nekretninama kao što su slike nekretnina, opis, veličina te lokacija ali i neke dodatne mogućnosti u vidu konkretnog tlocrta nekretnine na karti, procjene vrijednosti nekretnine, pregleda obližnjih ustanova itd.



Uz jednostavno i intuitivno sučelje Zillow se brzo pretvorio u vodeću platformu u sektoru nekretnina te se danas ističe količinom dostupnih podataka na stranici. Prikaz naslovne stranice Zillow može se vidjeti na slici 2.1.

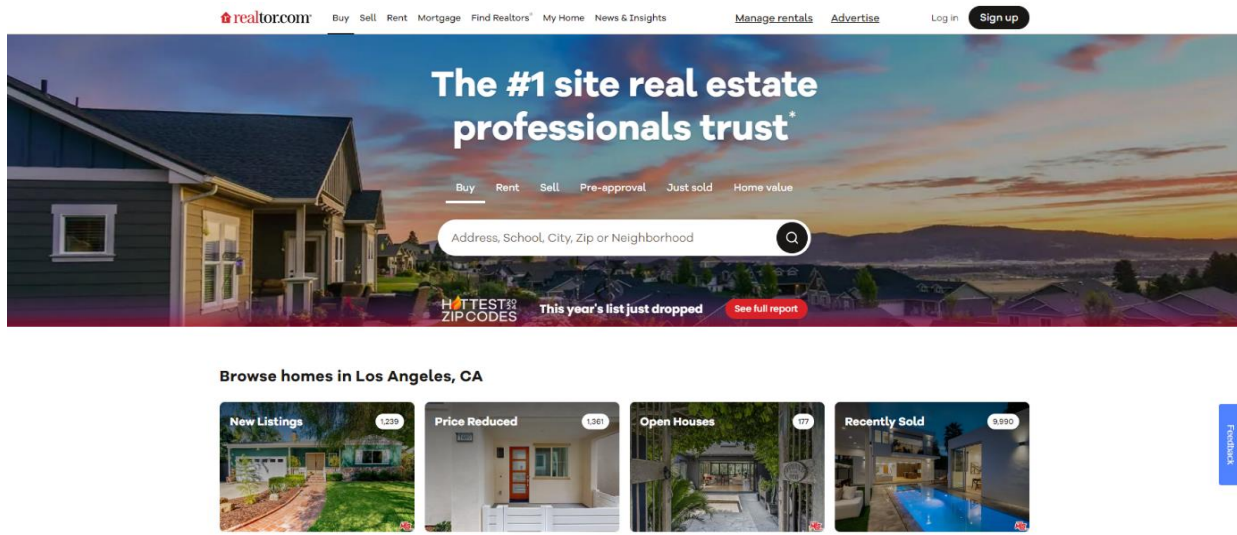


Sl. 2.1. Zillow naslovna stranica

## 2.2. Realtor.com

Za razliku od Zillow-a, Realtor.com je jedna od najstarijih i najpouzdanijih platformi za nekretnine u Sjedinjenim Amerčkim Državama, pokrenuta je 1995. godine. Budući da ona služi kao službena platforma Nacionalnog udruženja posrednika u prometu nekretninama, ova platforma posjeduje velik izbor nekretnina čiji pristup nudi i korisnicima. Dok Zillow nudi bogatiju ponudu različitih alata i širu bazu korisnika, Realtor.com se ističe od konkurencije zbog točnosti i preciznosti svojih informacija koju postižu svojom povezanošću s različitim listing servisima [2].

Kao i izrađena aplikacija, Realtor.com nudi uobičajenu pretragu nekretnina prema različitim kriterijima kao što su lokacija, cijena, veličina, tip nekretnine itd. Ova platforma još nudi korisnicima izravnu komunikaciju s agentima kako bi cjelokupan proces bio što lakši. Svaka nekretnina ima detaljan i iscrpan opis točno onoga što nudi: slike, cijenu, informacije, broj soba, kupatila itd. Realtor.com ima i mobilnu aplikaciju koja omogućuje pretragu nekretnina u pokretu. Također nudi čisto i profesionalno sučelje što se može vidjeti na slici 2.2. Na kraju, Realtor.com ostaje preferirana platforma za one koji cijene precizne i pouzdane informacije o nekretninama.

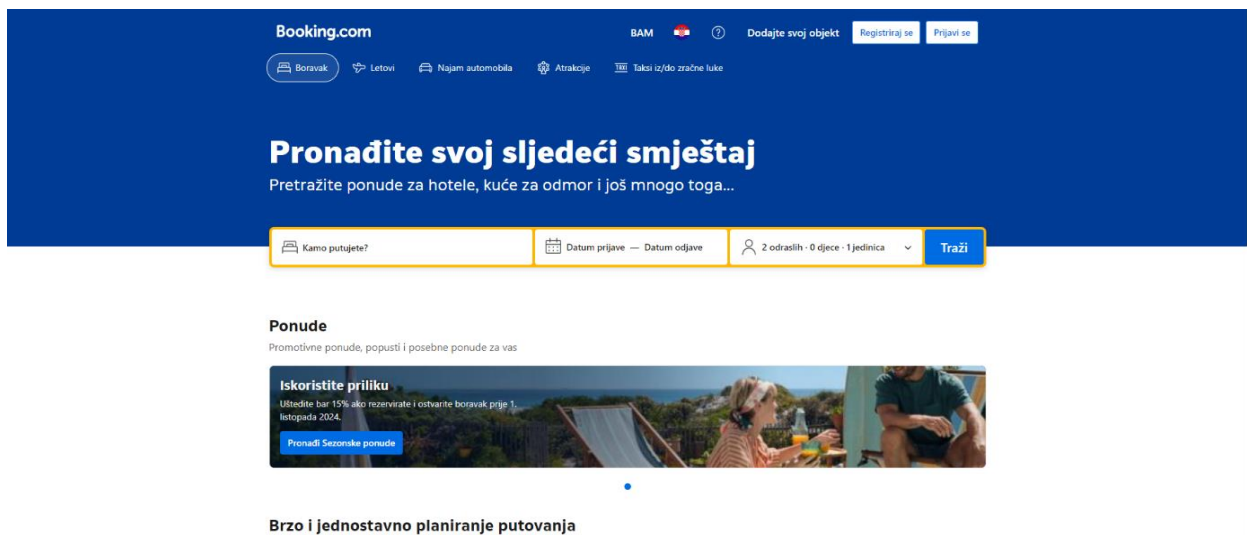


Sl. 2.2. Realtor.com naslovna stranica

## 2.3. Booking.com

Booking.com je jedna od najvećih i najpoznatijih online platformi za rezervaciju smještaja u svijetu. Osnovana je 1996. godine te je od tada postala sinonim za jednostavno i brzo pronalaženje hotela, apartmana, vila i drugih vrsta smještaja a danas nudi više od 28 milijuna smještajnih jedinica u preko 220 zemalja, omogućujući korisnicima lak pronalazak i rezervaciju smještaja koji odgovara njihovim potrebama [3].

Booking.com omogućuje korisnicima pretragu smještaja prema lokaciji, datumu dolaska i odlaska, broja gostiju itd. Na slici 2.3. može se vidjeti izgled naslovne stranice Booking.com. Za razliku od izrađene aplikacije, Booking.com nudi korisnicima samo pregled i rezervaciju smještaja bez mogućnosti prodaje nekretnina. No to ju i čini savršenim izborom za brži i efikasniji odabir smještaja na određeno razdoblje. Osim toga, korisnici mogu vidjeti i recenzije te ocjene drugih korisnika koji su tamo stanovali što im znatno olakšava odluku privremenog boravka. Booking.com također ima i mobilnu aplikaciju koja olakšava traženje smještaja u putu te dobivanje obavijesti o novim smještajima. U usporedbi s drugim sličnim platformama poput Airbnb, Booking.com nudi širi spektar smještaja i jednostavniji proces rezervacije. Uz to, Booking.com se ističe zbog značajki besplatnog otkazivanja smještaja te integriranog sustava za korisničku podršku. Besplatno otkazivanje nudi korisnicima fleksibilnost promjene planova a korisnička podrška otklanjanje problema s rezervacijama.

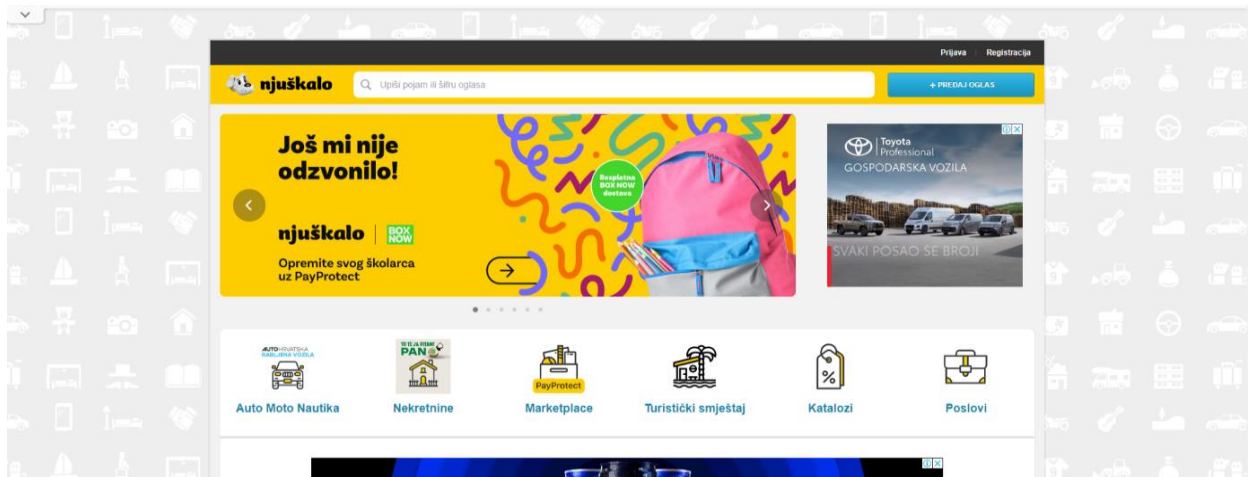


Sl. 2.3. Booking.com naslovna stranica

## 2.4. Njuškalo

Njuškalo je najveća hrvatska online platforma za oglašavanje i trgovinu, stoga naravno uključuje i nekretnine. Osnovana je 2007. godine te nalik na prethodno navedene platforme zbog jednostavnog sučelja brzo se proširila te je postala vodeće mjesto za kupoprodaju nekretnina u Hrvatskoj. Na slici 2.4. može se vidjeti izgled naslovne stranice Njuškala. Njuškalo nudi detaljno pretraživanje nekretnina uključujući filtre po cijeni, kvadraturi, broju soba, kupaonica, adresi, tipu nekretnine itd. Njuškalo također ima i mobilnu aplikaciju što omogućuje korisnicima pretraživanje oglasa, postavljanje oglasa i primanje obavijesti u pokretu. Njuškalo od izrađene aplikacije razlikuje opseg različitih tipova oglasa te nešto stariji oblik korisničkog sučelja. Njuškalo se fokusira samo na lokalno tržište Hrvatske i okolice i nudi jednostavniji i pristupačniji pristup oglašavanju [4].

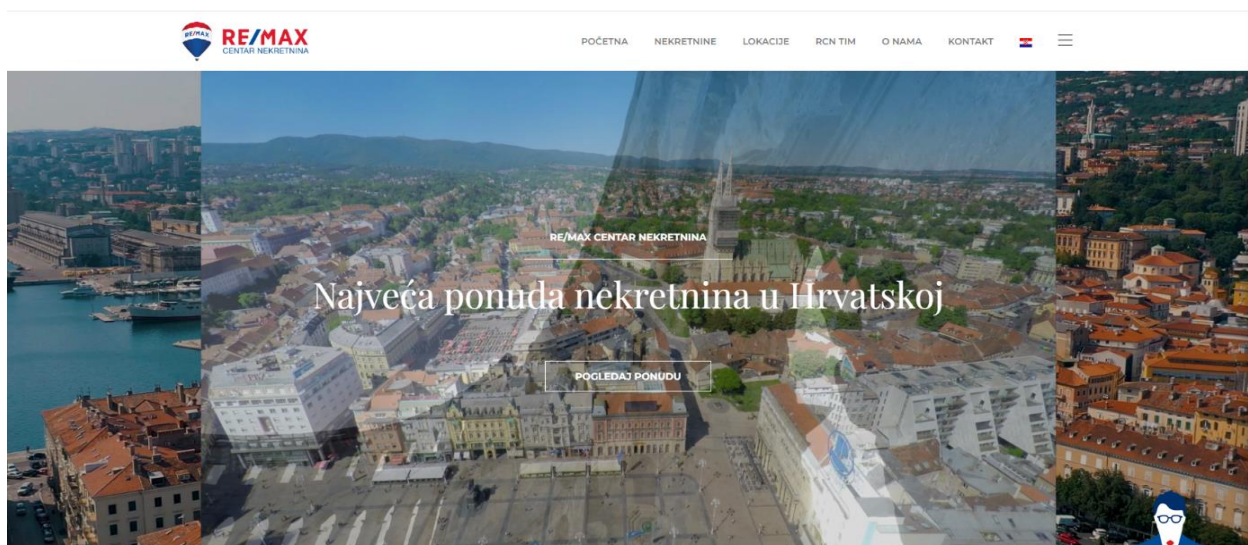
Jedna od ključnih značajki Njuškala je velik broj privatnih oglasa, što znači da korisnici mogu stupiti u kontakt s izdavačima ili prodavačima izravno, a ne preko nekih agencija te također korisnici mogu objaviti svoje vlastite oglase. Uz širok spektar oglasa koje Njuškalo izdaje dolaze i nedostaci poput manje relevantnih ili detaljnih informacija u oglasima te širok spektar oglasa, no Njuškalo i dalje ostaje vodeća opcija za veliki broj korisnika u Hrvatskoj.



Sl. 2.4. Njuškalo naslovna stranica

## 2.5. CentarNekretnina

CentarNekretnina je specijalizirana hrvatska web platforma za nekretnine koja se fokusira na pružanje detaljnih i profesionalnih oglasa za prodaju i najam nekretnina [5]. Ova platforma, kao i izrađena aplikacija, se fokusira na preciznost i pouzdanost informacija o nekretninama što se očituje u oglasima koji sadrže detaljne informacije o nekretninama uključujući kvalitetne fotografije, tlocrte, informacije o agentu ili agenciji koja izdaje oglas itd. CentarNekretnina omogućuje korisnicima pretraživanje nekretnina po različitim kriterijima poput cijene, kvadrature, lokacije itd. Svaka nekretnina ima stranicu s detaljnim informacijama uključujući slike, tip



Sl. 2.5. CentarNekretnina naslovna stranica

nekretnine itd. I CentarNekretnina ima mobilnu aplikaciju. CentarNekretnina također sadrži i alate za procjenu tržišne vrijednosti te kalkulatore hipoteka nekretnina. U usporedbi s Njuškalom nudi preciznije i detaljnije oglase koji su specijalizirani za nekretnine no zbog toga sadrži užu ponudu oglasa ciljanu samo za ozbiljne korisnike koji traže kvalitetne oglase. Izgled naslovne stranice CentarNekretnina prikazan je na slici 2.5.

### 3. ZAHTJEVI ZA APLIKACIJU

Razvoj weba započeo je s pojavom osnovnih tehnologija kao što su HTML, CSS i JavaScript. HTML (engl. *HyperText Markup Language*), koji je prvi put predstavljen 1991. godine je omogućio strukturiranje sadržaja na web stranicama. Nakon njega uslijedio je CSS (engl. *Cascading Style Sheets*), koji je osmišljen kako bi odvojio izgled odnosno stil stranice od njenog sadržaja te je tako omogućena bolja modularnost odnosno kontrola nad dizajnom. JavaScript je uveden 1995. godine te je omogućio dinamičnost i interaktivnost tako što donosi izradu funkcionalnih elemenata na web stranicama.

Budući da su navedene tehnologije samo osnove za izradu web stranica u modernom razdoblju došlo je do zahtjeva za sve složenijim web aplikacijama koje zahtijevaju puno više od samog prikaza sadržaja te se tako dolazi do podjele na backend i frontend tehnologije.

Pojam frontend razvoj se odnosi na sve ono što korisnik vidi (engl. *Client side*) i ono s čime izravno sudjeluje što se drugim riječima može nazvati sučeljem web aplikacije. Glavni cilj frontend razvojnog programera je osigurati intuitivnost, responzivnost i interaktivnost korisničkih sučelja. HTML, CSS i JavaScript spadaju u tu skupinu tehnologija, ali u modernijem razvoju web aplikacija frontend počinje obuhvaćati i složenije alate kao što su programski okviri (engl. *Framework*) i biblioteke (engl. *Libraries*) koje omogućuju izradu bogatih korisničkih sučelja. Često korišteni frontend alati koji pojednostavljuju proces razvoja tih složenih sučelja su React, Vue ili Angular.

Suprotno frontend razvoju nalazi se backend razvoj odnosno razvoj pozadinske funkcionalnosti (engl. *Server side*). Backend se bavi onime s čime se korisnik ne susreće te ga ne zanima što se odvija "iza kulisa". Drugim riječima backend je kod koji se izvršava na poslužiteljskoj strani, prima zahtjeve od klijenata i sadrži potrebnu logiku za slanje odgovarajućih podataka nazad klijentu. Tu spadaju one tehnologije koje se bave obradom podataka, upravljanjem bazama podataka, autentifikacijom korisnika te komunikacijom s frontendom kroz API-je (engl. *Application Programming Interface*). Shodno tome, glavne zadaće programera koji se bavi backendom su osiguravanje pravilne pohrane te obrađenosti svih podataka te učinkovit i siguran rad aplikacije. Često korištene backend tehnologije su Node.js, Express.js, te razni sustavi baza podataka poput MongoDB-a ili MySQL-a.

Ovakva podjela tehnologija na dva dijela omogućuje bolju učinkovitost i specijalizaciju prilikom razvoja budući da se programeri fokusiraju na manje određenih aspekata same aplikacije.

U konačnici, dobro usklađeni frontend i backend osiguravaju korisniku dobro korisničko iskustvo, ali i stabilnu i sigurnu funkcionalnost aplikacije.

### 3.1. HTML

HTML (engl. *HyperText Markup Language*) je opisni jezik korišten za izradu web stranica [6]. Pomoću hipertekstualne veze korisnicima omogućuje povezivanje s drugim dokumentima. Razvijen je početkom 1990-ih od strane Tima Berners-Leeja te njegovog tima znanstvenika u Ženevi. Prvi put je predstavljen javnosti 1991. godine kao jednostavan jezik koji služi za strukturiranje sadržaja na internetu. Kroz jednostavnu, ali vrlo učinkovitu sintaksu, HTML omogućuje organizaciju sadržaja poput teksta, slika, videozapisa i veza na internetu. Njegova glavna uloga je označavanje i definiranje elemenata na stranici, čineći ih razumljivima preglednicima, koji potom taj sadržaj prikazuju korisnicima. U povijesti razvoja weba, HTML se pokazao kao ključna tehnologija koja, unatoč svojoj jednostavnosti, omogućuje izgradnju složenih i interaktivnih web aplikacija. Kroz evoluciju različitih verzija, HTML je postajao sve moćniji, zadržavajući svoju jednostavnost, ali omogućujući sve naprednije funkcionalnosti u kombinaciji s drugim tehnologijama poput CSS-a i JavaScript-a, što ga čini neizostavnim dijelom modernog web razvoja.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap"
      rel="stylesheet"
    />
    <title>Nekretnine</title>
    <script
      src="https://upload-widget.cloudinary.com/global/all.js"
      type="text/javascript"
    >
    </script>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

Sl. 3.1. Primjer HTML koda

HTML oznake (engl. *tags*) ključni su elementi koji definiraju strukturu web stranice. Oznake se koriste za označavanje različitih dijelova sadržaja, a primjer oznaka te njihovog korištenja može se vidjeti na slici 3.1. Oznake u HTML-u mogu imati razne funkcionalnosti od kojih su najčešće:

- h1 - naslov najviše razine
- p - paragraf
- div - blok teksta
- span - inline element
- img - slika

Svaka oznaka obuhvaća sadržaj između početne i završne oznake, dajući preglednicima upute kako taj sadržaj prikazati. Osim osnovnih oznaka, HTML uključuje i semantičke oznake koje dodatno opisuju svrhu pojedinih dijelova stranice, olakšavajući čitanje i interpretaciju i ljudima i računalnim sustavima.

## 3.2. CSS

CSS (engl. *Cascading Style Sheets*) koristi se za stiliziranje web aplikacije [7]. Uveden je sredinom 1990-ih te je od tada postao neizostavan alat svakog dizajnera zato što omogućuje stvaranje vizualno privlačnih komponenti na jednostavan način primjenom određenih svojstava komponentama HTML koda. Primjenjuje se na HTML elemente preko klasifikatora unutar oznaka elemenata i definira kako će oni biti prikazani na stranici. Pruža puno veću fleksibilnost u odnosu na čisti HTML te je moguće definirati boju fonta i pozadine, nijanse, prozirnost itd. Prednosti CSS-a su što je lagan za naučiti i razumjeti te modularan i ne predstavlja problem razvojnim programerima u smislu razumijevanja koda.

Moguće je primijeniti CSS na više načina: pomoću vanjske primjene liste stilova (engl. *external stylesheet*), pomoću unutarnje liste stilova (engl. *internal stylesheet*) te pomoću stila unutar linije (engl. *inline style*). Najčešće se zbog spomenute modularnosti koda bira vanjska primjena liste stilova te će se ona implementirati i u web aplikaciji Nekretnine preko posebnog dokumenta koji



se zove *Komponenta.css* koji se uvozi (engl. *import*) u dokumente različitih komponenti aplikacije. Logo CSS-a moguće je vidjeti na slici 3.3.



Sl. 3.2. CSS logo

Prilikom pisanja CSS stilova potrebno je koristiti određenu sintaksu: oznaka { svojstvo1: vrijednost1; svojstvo2 : vrijednost2 }. Primjer CSS koda moguće je vidjeti na slici 3.4.

```

:root {
  --primary: #1f3e72;
  --secondary: rgba(255, 255, 255, 0.78);
  --black: #131110;
  --blue-gradient: linear-gradient(97.05deg, #4066ff 3.76%, #2949c6 100%);
  --orange-gradient: linear-gradient(270deg, #ffb978 0%, #ff922d 100%);
  --blue: #4066ff;
  --lightBlue: #eeeeff;
  --shadow: 0px 23px 21px -8px rgba(136, 160, 255, 0.25);
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
  scroll-behavior: smooth;
}

.wrapper {
  background-color: white;
}

a {
  color: inherit;
  text-decoration: none;
}

.paddings {
  padding: 1.5rem;
}

.innerWidth {
  width: 100%;
}

.flexCenter {
  display: flex;
  row-gap: 2rem;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}

```

Sl. 3.3. Primjer CSS koda

### 3.3. MERN

MERN je akronim za skup tehnologija koje se koriste za razvoj modernih web aplikacija [8]. To su tehnologije koje zajedno omogućuju razvojnim programerima razvoj sveobuhvatnih web aplikacija koristeći JavaScript. MERN stog uključuje sljedeće tehnologije:

1. MongoDB
2. Express.js
3. React
4. Node.js

U MERN stogu React se koristi u frontendu za razvoj korisničkog sučelja aplikacije. React sa svojim komponentama omogućuje programerima stvaranje dinamičnih i interaktivnih elemenata koji se učitavaju u pregledniku korisnika. Za backend razvoj web aplikacije koristi se Express.js i Node.js. Express.js omogućuje stvaranje krajnjih točaka koje frontend koristi za komunikaciju s

backendom te manipulaciju podacima. Krajnje, za bazu podataka koristi se MongoDB koji pohranjuje podatke koje aplikacija koristi u BSON (engl. *Binary JavaScript Object Notation*) formatu, unutar MongoDB dokumenata.

MERN stog pruža razvojnim programerima mnoge prednosti. Jedna od njih je jedinstveni jezik JavaScript kojega svi dijelovi stoga koriste što omogućava dosljednost te uvelike olakšava rad razvojnim programerima. Kombinacija React-a za frontend i Express.js za backend omogućuje brzu izgradnju aplikacija s bogatim korisničkim sučeljem i efikasnim upravljanjem podacima. MongoDB pruža fleksibilnost u modeliranju podataka dok Node.js omogućuje skaliranje aplikacije za velik broj korisnika i zahtjeva.

MERN stog je popularan izbor za razvoj modernih web aplikacija zbog svoje sposobnosti pokrivanja svih aspekata razvoja aplikacije – od korisničkog sučelja do backend logike i pohrane podataka.

### **3.4. MongoDB**

MongoDB je jedna od najpopularnijih NoSQL baza podataka koja se koristi za pohranu podataka u obliku dokumenata te pruža željenu fleksibilnost i skalabilnost [9]. Za razliku od tradicionalnih relacijskih baza podataka, koje koriste tablice za organizaciju podataka, MongoDB pohranjuje podatke u fleksibilnim BSON dokumentima što znači da se polja mogu razlikovati od dokumenta do dokumenta, a struktura podataka može se mijenjati tijekom vremena odnosno omogućava pohranu i upravljanje nestrukturiranim ili polustrukturiranim podacima na učinkovit način.

Zahtjevi prema bazi podataka uključuju sigurno i efikasno pohranjivanje podataka, fleksibilnost u rukovanju različitim tipovima podataka te mogućnost skaliranja kako bi podržala veći broj korisnika ili povećan volumen podataka. Model dokumenata se mapira na objekte u aplikacijskom kodu, što čini rad s podacima jednostavnim. Također je besplatna i otvorenog koda te je razvijena 2009. godine od strane istoimene kompanije.

U kontekstu MERN (MongoDB, Express.js, React, Node.js) stoga, MongoDB pruža solidan temelj za pohranu podataka, omogućujući brzu i pouzdanu komunikaciju s ostalim komponentama aplikacije.

### 3.5. Express.js

Minimalistički, brz i fleksibilan web programski okvir za Node.js, koji olakšava izgradnju i upravljanje backend poslužiteljima koristeći JavaScript [10]. Predstavlja standard poslužiteljskog programskog okvira za Node.js. Njegova jednostavnost, fleksibilnost i bogata zajednica predstavljaju ga jednom od najboljih opcija za moderne web projekte.

Express.js omogućuje brzu izradu API-a i rješavanje HTTP zahtjeva i odgovora. Jedna od velikih prednosti Express.js-a je njegovo rukovanje rutama. Routing omogućuje definiranje različitih URL (engl. *Uniform Resource Locator*) putanja unutar aplikacije te određivanje odgovora aplikacije na različite HTTP metode (GET, POST, PUT, DELETE).

U backend zahtjeve prema aplikaciji na poslužiteljskoj strani aplikacija koristi Node.js kao okruženje i Express.js kao web okvir za izradu REST API-ja. Zahtjevi prema backendu uključuju obradu zahtjeva korisnika, sigurnu autentifikaciju, učinkovito upravljanje sesijama, te osiguranje brzih odgovora iz baze podataka.

### 3.6. React

ReactJS, odnosno React, je popularna JavaScript biblioteka za izgradnju korisničkih sučelja [11]. Također ju se oslovljava kao frontend JavaScript biblioteka. Razvio ga je Facebook nakon što su otkrili sporost tradicionalnog DOM-a (engl. *Document Object Model*) i široko se koristi za stvaranje dinamičnih i interaktivnih web aplikacija [12].

React je riješio taj problem sporosti i brzo stekao popularnost te se i dalje razvija stalno uvodeći nove značajke. React je deklarativna knjižnica temeljena na komponentama koja omogućava razvojnim programerima izradu ponovno upotrebljivih UI (engl. *User Interface*) komponenti [13]. Pomoću manjih komponenti koje su neovisne i više puta se koriste na različitim mjestima dobiva se složenost aplikacije. Svaka od tih komponenti ima svoje stanje (engl. *state*), svojstva (engl. *props*), kontekst (engl. *context*) i efekt (engl. *effect*). Sve komponente su povezane jednosmjerno na način da se podaci prenose od roditeljskih komponenti prema dječjim komponentama te preko svojstava u dječjim komponentama mogu prilagoditi stanje prema pruženim unosima.

React koristi pristup Virtual DOM koji optimizira performanse renderiranja minimiziranjem ažuriranja DOM-a. Također je brz i dobro radi s drugim alatima i knjižnicama. Još jedna prednost korištenja Reacta je JSX (engl. *JavaScript Syntax Extension*), odnosno pisanje koda nalik HTML-u unutar JavaScript datoteka što čini komponente čitljivijima i izražajnijima.

Aplikacija koristi React za razvoj dinamičkog korisničkog sučelja. Ovdje su zahtjevi definirani u smislu performansi (učinkovito renderiranje komponenti, responzivnost), funkcionalnosti (interaktivnost, routing), i modularnosti koda (komponente).

### **3.7. Node.js**

Node.js je otvorenog koda (engl. *open-source*) i višepatformsko (engl. *cross-platform*) JavaScript okruženje za izvršavanje te popularan alat za gotovo svaki projekt [14]. Također omogućuje izvođenje JavaScript koda izvan preglednika. Node.js se koristi za izgradnju aplikacija s poslužiteljske strane. Aplikacije u Node.js-u rade u jednom procesu, bez kreiranja novih niti za svaki zahtjev. Node.js omogućava rad s asinkronim I/O operacijama što mu omogućuje obradu tisuća istodobnih veza s jednim poslužiteljem bez uvođenja opterećenja upravljanja konkurencijom niti, što bi moglo biti značajan izvor grešaka.

Node.js ima jedinstvenu prednost jer milijuni frontend razvojnih programera koji pišu JavaScript za preglednike sada mogu pisati kod s poslužiteljske strane uz kod s klijentske strane, bez potrebe za učenjem potpuno drugačijeg jezika.

U Node.js-u se novi ECMAScript standardi mogu koristiti bez problema, zato što se ne mora čekati da svi korisnici ažuriraju svoje preglednike – razvojni programer je taj koji odlučuje koju verziju ECMAScripta koristiti promjenom verzije Node.js-a, a također može omogućiti specifične eksperimentalne značajke pokretanjem Node.js-a s opcijama.

## 4. POSTUPAK IZRADE APLIKACIJE

U ovome poglavlju je opisan postupak izrade web aplikacije za nekretnine te njezine funkcionalnosti za koje koristi prije opisane tehnologije. Aplikacija se sastoji od više elemenata a oni su baza podataka, pozadinska funkcionalnost (backend usluge) te prikaz podataka na korisničkom sučelju (frontend). Aplikacija omogućuje svojim korisnicima pregled svojeg sadržaja odnosno nekretnina ako nisu prijavljeni, te registraciju i prijavu nakon čega dobivaju mogućnost dodavanja svojih nekretnina za prodaju te prijavu za pregled drugih nekretnina i dodavanje određenih nekretnina među omiljene.

### 4.1. Baza podataka

Baza podataka je organizirani skup podataka koji se pohranjuje i upravlja na strukturiran način, omogućujući jednostavno pretraživanje, ažuriranje i manipulaciju podacima. Na samome početku izrade web aplikacije potrebno je osmisliti kakve korisnike će baza koristiti, koliko će ih biti te s čime i kako će oni sudjelovati.

Za bazu web aplikacije je korišten MongoDB a on nudi određene fleksibilnosti prilikom rada s bazom podataka zato što ne traži strogo strukturirane dokumente za rad. U ovome radu također je korišten alat za objektno-relacijsko mapiranje (engl. *Object-Relational Mapping*) Prisma koji korisnicima omogućuje korištenje JavaScript jezika prilikom komunikacije s bazama podataka, automatizaciju generiranja upita itd. Prisma predstavlja generalni način za rukovanje bazama podataka apstrakciju baze podataka te tako olakšava razvojnim programerima rad s bazom podataka.

Na slici 4.1. se može vidjeti prikaz koda u programu VisualStudio Code koji je korišten odnosno prisma kod koji opisuje modele baze podataka te veze između njih [15]. Postoje dva modela u bazi podataka a oni su korisnik (engl. *User*) te nekretnina (engl. *Residency*). Svaki od njih imaju svoje atribute. Za korisnika to su atributi koji predstavljaju njegov identifikacijski niz znakova koji se sam automatski postavlja na određenu vrijednost te sprema u bazu podataka kao takav, puno ime korisnika, e-mail koji mora biti jedinstven, sliku, polje posjeta koje je dogovorio predstavljeno poljem JSON elemenata, polje najdražih nekretnina te polje nekretnina čiji je vlasnik korisnik koje je predstavljeno poljem elemenata nekretnina a predstavlja relaciju na model nekretnine. Korisnici aplikacije mogu biti kupci kojima se nakon prijave otvaraju mogućnosti rezervacije posjete nekretninama te dodavanje nekretnina u omiljene te prodavači koji uz navedene mogućnosti mogu i dodati svoje nekretnine za prodaju. Drugi model u bazi podataka je model

nekretnina koji također ima neke slične attribute kao što su identifikacijski niz znakova, naslov, opis, cijena, slika ali ima još i neke druge. To su atributi vezani za lokaciju nekretnine koji će biti iskorišteni kako bi korisnici mogli na karti vidjeti gdje se nekretnina nalazi te vlasnik koji predstavlja korisnika koji je objavio nekretninu te ima jedinstvena polja za e-mail koja odgovaraju vlasniku koji je objavio svoju nekretninu. Još ima i attribute koji opisuju kada je vlasnik objavio odnosno uredio svoju nekretninu.

```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mongodb"
  url      = env("DATABASE_URL")
}

model User {
  id          String      @id @default(auto()) @map("_id") @db.ObjectId
  name       String?
  email      String      @unique
  image      String?
  bookedVisits  Json[]
  favResidenciesID String[] @db.ObjectId
  ownedResidencies Residency[] @relation("Owner")
}

model Residency {
  id          String      @id @default(auto()) @map("_id") @db.ObjectId
  title      String
  description String
  price      Int
  address    String
  city       String
  country    String
  image      String
  facilities  Json
  userEmail  String
  owner      User        @relation("Owner", fields: [userEmail], references: [email])
  createdAt  DateTime @default(now())
  updatedAt  DateTime @default(now())

  @@unique(fields: [address, userEmail])
}
```

Sl. 4.1. Prikaz koda modela

Na slici 4.2. se može vidjeti prikaz s MongoDB platforme na kojemu se vidi baza podataka u kojoj se nalaze spremljeni korisnici sa svim svojim atributima u odgovarajućem BSON formatu. Također vidi se postojanje nekretnina spremljenih na identičan način.

Sl. 4.2. Prikaz podataka spremljenih u bazu

## 4.2. Komponente

U Reactu, komponente su osnovni gradivni elementi koji omogućuju izgradnju korisničkog sučelja. Komponente mogu biti jednostavne funkcionalne komponente ili složenije klase. One su modularne i omogućuju ponovnu upotrebu koda, što olakšava razvoj i održavanje aplikacija. Svaka komponenta može imati svoje vlastito stanje (engl. *state*) i svojstva (engl. *props*), što joj omogućuje dinamičko ponašanje i prilagodljivost korisničkom unosu. Na slici 4.3. je prikazan kod glavnog dijela aplikacije u kojem se koristi tzv. React kuka (engl. *hook*) naziva *useState* te služi kao spremnik za detalje o korisniku te poziv drugih stranica putem zadanih putanja. Također se mogu vidjeti komponente koje čine samu aplikaciju kao što su:

- Layout – osnovna putanja koja se primjenjuje na sve druge komponente aplikacije a predstavlja raspored elemenata na prikazima aplikacije
- Website – samozatvarajuća oznaka koja se poziva prilikom samog ulaska u aplikaciju te označava naslovnicu aplikacije
- Properties – prikaz svih raspoloživih nekretnina
- Property – prikaz informacija o jednoj nekretnini
- Bookings – prikaz nekretnina za koje je korisnik zakazao pregled nekretnine
- Favourites – prikaz nekretnina koje je korisnik označio kao omiljene

Na slici 4.3. može se vidjeti prikaz korištenja načina pisanja koda specifičnog za React koji se naziva JSX i veći broj komponenti koje "omotavaju" srž aplikacije kako bi ona mogla raditi na



```

function App() {
  const queryClient = new QueryClient();
  const [userDetails, setUserDetails] = useState({
    favourites: [],
    bookings: [],
    token: null
  })
  return (
    <UserDetailContext.Provider value={{ userDetails, setUserDetails }}>
      <QueryClientProvider client={queryClient}>
        <BrowserRouter>
          <Suspense fallback={<div>Loading...</div>}>
            <Routes>
              <Route element={<Layout />}>
                <Route path="/" element={<Website />} />
                <Route path="/properties">
                  <Route index element={<Properties />} />
                  <Route path=":propertyId" element={<Property />} />
                </Route>
                <Route path="/bookings" element={<Bookings />} />
                <Route path="/favourites" element={<Favourites />} />
              </Route>
            </Routes>
          </Suspense>
        </BrowserRouter>
        <ToastContainer />
        <ReactQueryDevtools initialIsOpen={false} />
      </QueryClientProvider>
    </UserDetailContext.Provider>
  );
}

export default App;

```

Sl. 4.3. Prikaz glavnog dijela aplikacije

željeni način [16]. To su UserDetailContext, QueryClientProvider, BrowserRouter i Suspense čija je uloga pružati informacije o klijentu i rad s putanjama.

Na slici 4.4. je prikazan kod složene komponente Property koja predstavlja jedinstvenu nekretninu a prilikom pisanja navedenog koda korišten je formater koda Prettier [17]. Komponenta Property prvo koristi jednostavnu komponentu Heart koja prikazuje oblik srca na desnom gornjem kutu slike nekretnine na koji se može kliknuti kako bi korisnik dodao nekretninu među svoje omiljene nekretnine. Potom se u kodu poziva naziv nekretnine te cijena što se može vidjeti na slici 5.2. Također koristi neke ugrađene React komponente kao što su FaShower, AiTwotoneCar te

```

const Property = () => {
  const { pathname } = useLocation();
  const id = pathname.split("/").slice(-1)[0];
  const { data, isLoading, isError } = useQuery(["residency", id], () => getProperty(id)
  );
  const [modalOpened, setModalOpened] = useState(false)
  const validateLogin = useAuthCheck()
  const { user } = useAuth0();
  const { ...
} = useContext(UserDetailContext);
const { mutate: cancelBooking, isLoading: cancelling } = useMutation({...
})
if (isLoading) {...
}
if (isError) {...
}
return (
  <div className="wrapper">
    <div className="flexColStart paddings innerWidth property-container">
      <div className="flexColStart left">
        <div className="flexStart head">
          <span className='primaryText'>{data?.title}</span>
          <span className='orangeText' style={{ fontSize: "1.5rem" }}>${data?.price}</span>
        </div>
        <div className="flexStart facilities">
          <div className="flexStart facility">
            <FaShower size={20} color="#1F3E72" />
            <span>{data?.facilities?.bathrooms} Bathrooms</span>
          </div>
          <div className="flexStart facility">
            <AiTwotoneCar size={20} color="#1F3E72" />
            <span>{data?.facilities.parkings} Parking</span>
          </div>
          <div className="flexStart facility">
            <MdMeetingRoom size={20} color="#1F3E72" />
            <span>{data?.facilities.bedrooms} Room</span>
          </div>
        </div>
        <span className="secondaryText" style={{ textAlign: "justify" }}>{data?.description}</span>
        <div className="flexStart" style={{ gap: "1rem" }}>
          <MdLocationPin size={25} />
          <span className="secondaryText"> {data?.address}{" "} {data?.city}{" "} {data?.country}{" "} </span>
        </div>
        {bookings?.map((booking) => booking.id).includes(id) ? (
          <div>
            <Button variant="outline" w={"100%"} color="red" onClick={() => cancelBooking()} disabled={cancelling}>
              <span>Cancel booking</span> </Button>
            <span>Your visit booked for date {bookings?.filter((booking) => booking?.id === id)[0].date}</span>
          </div>
        ) : (
          <button
            className="button"
            onClick={() => { if (validateLogin()) { setModalOpened(true); }}}
            > Book your visit </button>
        )}
        <BookingModal opened={modalOpened} setOpened={setModalOpened} propertyId={id} email={user?.email} />
      </div>
    <div className="map"> <Map address={data?.address} city={data?.city} country={data?.country} /> </div>
  </div>
);
};
export default Property

```

Sl. 4.4. Prikaz koda komponente Property

MdMeetingRoom za prikaz koliko nekretnina ima toaleta, parking mjesta te soba. Za prikaz adrese koristi se također React ugrađena ikona igle kraj adrese. Još koristi i složenu korisničku BookingModal komponentu koja je prikazana na slici 5.3. te predstavlja gumb za zakazivanje pregleda nekretnine odnosno otkazivanje ako korisnik već ima zakazan pregled uživo. Korisnik nema mogućnost odabira datuma koji je prošao a nakon što zakaže pregled nekretnine moguće ga je i otkazati preko gumba koji se vidi na slici 5.4. Zadnja složena komponenta je Map i prikazana

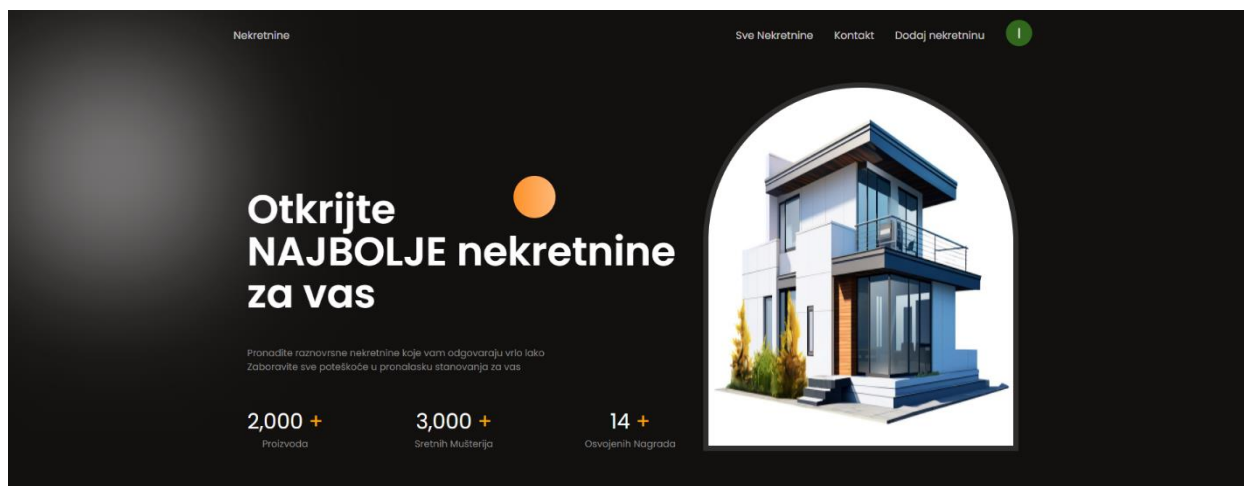
je također na slici 5.2. a služi za prikaz interaktivne karte s lokacijom nekretnine. Korisnik ima mogućnost zumiranja karte na željenu veličinu.

## 5. PRIKAZ RADA APLIKACIJE I KORISNIČKOG SUČELJA

U ovom poglavlju opisano je korisničko iskustvo od dolaska na naslovnicu, prijave korisnika, označavanja omiljenih nekretnina i zakazivanja pregleda nekretnine do postavljanja nekretnina.

### 5.1. Naslovnica

Prilikom ulaska na naslovnicu prikazano je više stvari: zaglavlje i podnožje koji predstavljaju osnovu svake stranice te se nalaze u Layout komponenti i implementiraju navigaciju po web aplikaciji. Klikom na naslov Nekretnine korisnik se vraće na naslovnicu web aplikacije čija je osnovna putanja samo "/". Na slici 5.1. prikazan je izgled naslovnice web aplikacije nakon što se korisnik prijavio. Prijavljeni korisnici imaju mogućnost dodavanja nekretnina te označavanje nekretnina kao omiljene kao i bilježenje posjeta nekretninama dok neprijavljeni korisnici mogu samo pregledati skup nekretnina te opciju kontakta čime se otvara e-mail servis pomoću kojega klijent može doći do voditelja stranice.



Sl. 5.1. Naslovnica aplikacije

### 5.2. Nekretnina

Na stranicu za prikaz pojedine nekretnine korisnik dolazi preko putanje "/properties/:propertyId" što znači da mu je potreban i identifikacijski niz znakova za pristup pojedinoj nekretnini, a njega najlakše dobije putem samih mogućnosti web aplikacije odnosno klikom na sliku određene nekretnine iz prikaza svih nekretnina. Kao i na svakoj drugoj stranici nalazi se zaglavlje na vrhu te podnožje na dnu stranice. Između se nalazi srž koja se na stranici nekretnine sastoji od slike nekretnine ispod koje su ime nekretnine, cijena, opis, lokacija, gumb za

## Plavi Horizont Villa

\$ 2500

2 Kupaonica Parking mjesta 3 Soba

Ogromna prilika za izgradnju vašeg doma iz snova u podnožju planine Mummy, u školskom okrugu 3 C's. Kuća je trenutno renovirana i vrlo pogodna za život, ako planirate gradnju kasnije. Bonus je što možete živjeti na padini bez ograničenja za padinsku gradnju, u gradu Paradise Valley. Ne čekajte, iskoristite ovu jedinstvenu priliku!

Street 13 Delhi India

Rezervirajte posjetu



### Nekretnine

Naša vizija je učiniti da svi ljudi žive u najboljem mjestu za njih.

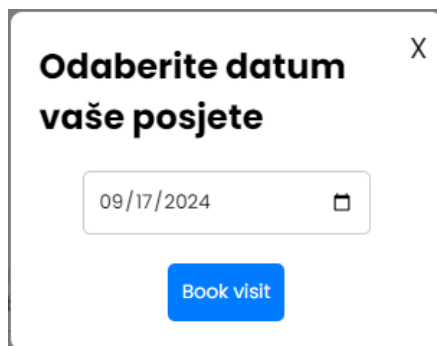
### Informacije

Frankopanska 92, 31000 Osijek

Ivan Koturić FERIT

### Sl. 5.2. Prikaz podataka jedne nekretnine

zakazivanje posjeta nekretnini ili otkazivanje ako je posjeta već zakazana te s desne strane interaktivna karta koja korisniku omogućuje pregled lokacije nekretnine. Klikom na gumb *Rezervirajte posjetu* otvara se dijaloški okvir za rezervaciju posjete nekretnini gdje korisnik unosi



### Sl. 5.3. Dijaloški okvir za zakazvanje posjete

## Summerhill Imanje

\$ 2000

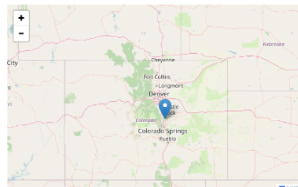
2 Kupaonica Parking mjesto 2 Soba

Ogromna prilika za izgradnju vašeg doma iz snova u podnožju planine Mummy, u školskom okrugu 3 C's. Kuća je trenutno renovirana i vrlo pogodna za život, ako planirate gradnju kasnije. Bonus je što možete živjeti na padini bez ograničenja za padinsku gradnju, u gradu Paradise Valley. Ne čekajte, iskoristite ovu jedinstvenu priliku!

Street 8 Denver US

Odjavi rezervaciju

Posjeta rezervirana za datum: 31/08/2024



### Nekretnine

Naša vizija je učiniti da svi ljudi žive u najboljem mjestu za njih.

### Informacije

Frankopanska 92, 31000 Osijek

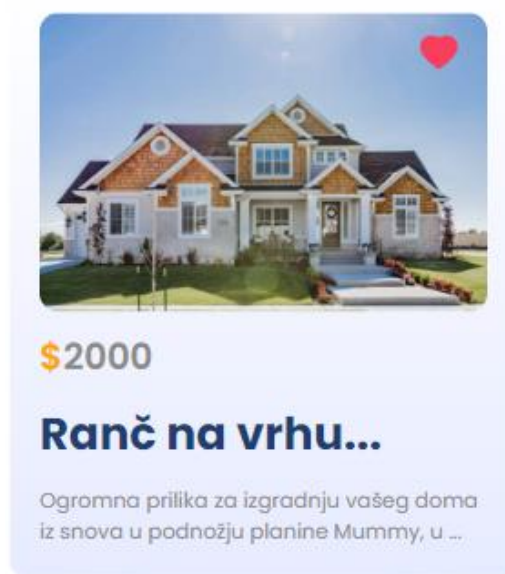
Ivan Koturić FERIT

### Sl. 5.4. Izgled prikaza o nekretnini nakon zakaze posjete

željeni datum posjete nekretnini. Nakon rezervirane posjete nekretnini na prikazu informacija može se vidjeti navedeni datum te gumb *Odjavi rezervaciju*.

### 5.3. Nekretnine

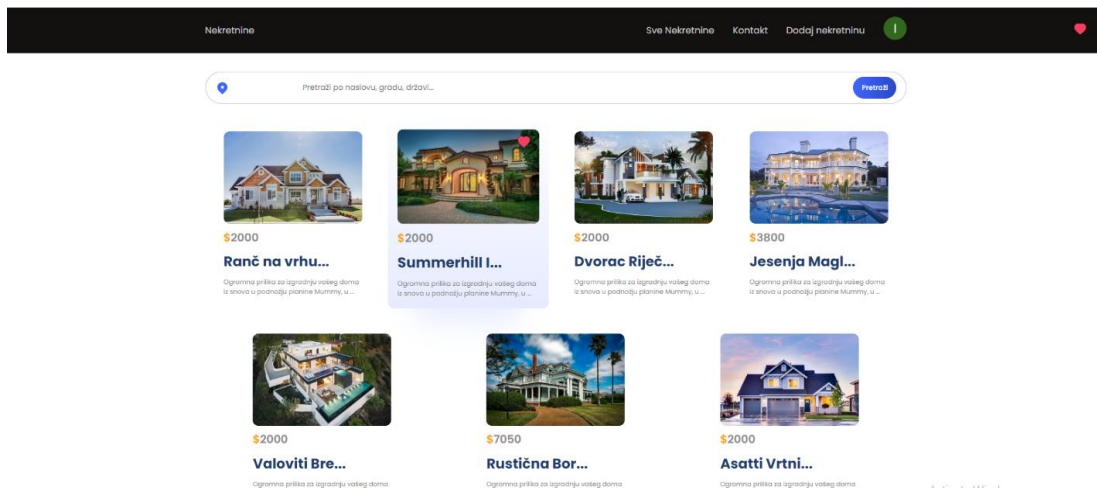
Na prikazu Nekretnine se inicijalno prikazuju sve nekretnine u istom formatu kao na slici 5.6. Postoji i traka za pretraživanje u koju korisnici mogu unijeti podatke o imenu, gradu ili državi kako bi filtrirali nekretnine koje ih zanimaju. S ovog prikaza moguće je označiti nekretninu kao omiljenu klikom na ikonu srca koja se pojavi prelaskom miša preko slike nekretnine što je prikazano na slici 5.5.



Sl. 5.5. Prikaz označene nekretnine

### 5.4. Omiljene nekretnine i Rezervacije

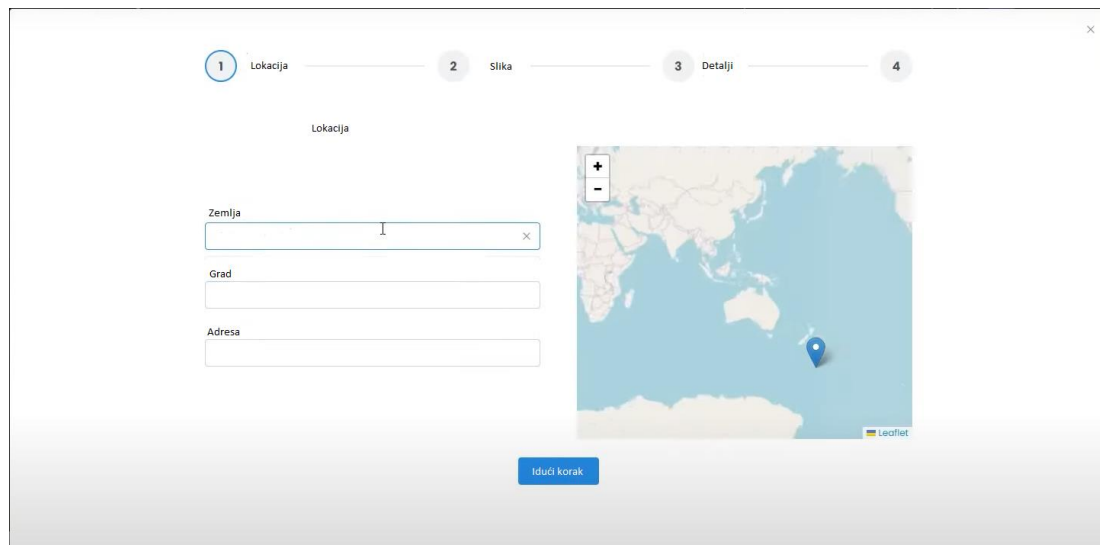
Stranice Omiljene nekretnine i Rezervacije koriste se za prikaz nekretnina koje je korisnik dodao u omiljene odnosno zakazao pregled nekretnine za kupovinu. Ako korisnik proba pristupiti stranici a nije prijavljen iskače toast obavijest s porukom da se korisnik prvo mora prijaviti. Klikom na svoju ikonu iskače dijaloški okvir za prelazak na druge stranice kao što se može vidjeti na slici 5.6.



Sl. 5.6. Prikaz stranice Omiljene nekretnine

## 5.5. Dodavanje nekretnina

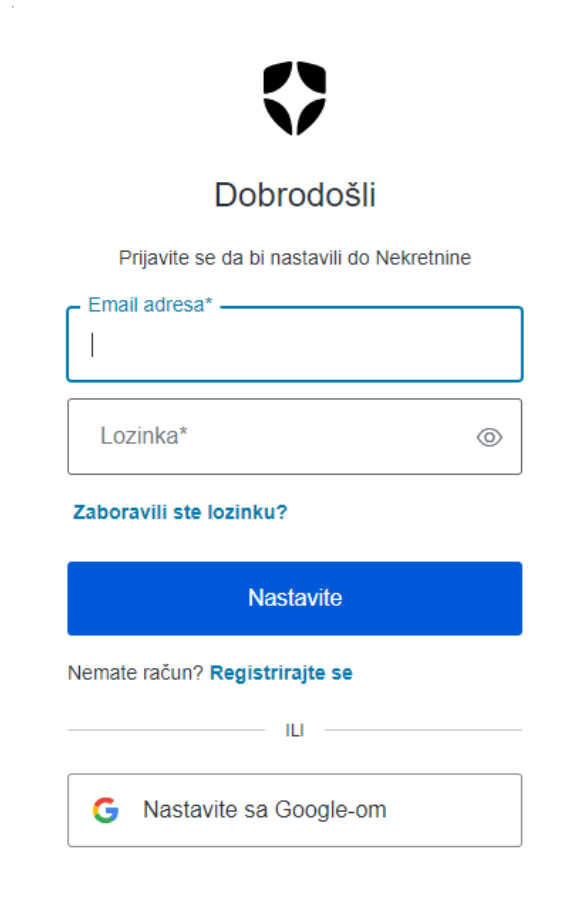
Korisnici koji su se prijavili imaju mogućnost dodavanja nekretnina klikom na gumb *Dodaj nekretninu* nakon čega se poziva komponenta *AddPropertyModal* čija je zadaća pozivanje dijaloškog okvira koji korisniku nudi određene opcije. Dijaloški okvir može se vidjeti na slici 5.7. Prvo je potrebno unijeti lokaciju nekretnine za koju se provjerava postoji li već u kojem slučaju nije moguće nastaviti. Drugo se dodaje slika koju korisnik može unijeti sa svojega računala ili interneta te potom detalji o nekretnini.



Sl. 5.7. Dijaloški okvir za dodavanje nekretnine

## 5.6. Autentifikacija i registracija korisnika

Budući da je autentifikacija u moderno doba jedna od najvažnijih stvari prilikom izrade web aplikacija kao i mnogih drugih projekata potrebno je posebno obratiti pozornost na njenu implementaciju.



Sl. 5.8. Prikaz prijave korisnika

Danas postoji veliki broj paketa koji se mogu koristiti kako bi autentifikacija korisnika bila brza i sigurna. Jedan od njih koji je korišten u ovoj aplikaciji zove se Auth0. Auth0 je platforma za autentifikaciju i autorizaciju korisnika koja omogućuje razvojnim programerima jednostavnu implementaciju složenih problema autentifikacije, integrira se s različitim programskim okvirima te pruža visoku razinu sigurnosti i fleksibilnosti. Na slici **Error! Reference source not found.** može se vidjeti stranica za prijavu te registraciju korisnika u bazu podataka koja se poziva prilikom klika na *Prijavi se* gumb. Prilikom registracije korisnik mora osigurati da lozinka koju unese ima dovoljan broj znakova, velikih i malih slova te najmanje jedan dijakritički znak. Auth0 nudi mogućnosti prijave na web aplikaciju putem unošenja podataka vlastoručno ili preko nekih drugih odabranih servisa kao što su Google, Apple itd.



## 6. ZAKLJUČAK

U moderno doba kreiranje web aplikacija postaje sve lakše, brže i efikasnije. Naravno, potrebno je poznavanje područja rada odnosno tehnologija koje su potrebne za izradu. U ovom završnom radu obrađena je web aplikacija za nekretnine te alati i vještine potrebne za njenu izradu kao i proces izrade. Za to je korišten MERN stog odnosno MongoDB, Express.js, React i Node.js uz HTML te CSS. Korisnicima je pruženo odlično korisničko iskustvo te korisničko sučelje što je u moderno doba jedna od najvažnijih stavki za zadržavanje novih korisnika. Korisnici imaju mogućnost pregleda dostupnih nekretnina te pregleda nekretnina uživo. Cilj aplikacije je naglasiti korisnost novih tehnologija poput React-a te lakoću izrade web aplikacija korištenjem istih. Aplikacija ostavlja i mjesta za napredak s obzirom da je izuzetno lako dodati nove komponente i funkcionalnosti.

## LITERATURA

- [1] Zillow, dostupno na: <https://www.zillow.com/>, [16.9.2024.]
- [2] Realtor.com, dostupno na: <https://www.realtor.com/>, [16.9.2024.]
- [3] Booking.com, dostupno na: <https://www.booking.com/>, [16.9.2024.]
- [4] Njuškalo, dostupno na: <https://www.njuskalo.hr/>, [16.9.2024.]
- [5] CentarNekretnina, dostupno na: <https://www.remax-centarnekretnina.com/>, [16.9.2024.]
- [6] HTML, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTML>, [16.9.2024.]
- [7] CSS, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/CSS>, [16.9.2024.]
- [8] MERN, dostupno na:  
<https://www.mongodb.com/resources/languages/mern-stack>, [16.9.2024.]
- [9] MongoDB, dostupno na: <https://www.mongodb.com/company/what-is-mongodb>, [16.9.2024.]
- [10] Express.js, dostupno na: <https://expressjs.com/>, [16.9.2024.]
- [11] K. Peterson, „React js: The Ultimate Beginner's Guide to Learn React js Programming Step by Step – 2020“, mEm Inc
- [12] react-router-dom, dostupno na: <https://www.npmjs.com/package/react-router-dom>, [16.9.2024.]
- [13] React, dostupno na: <https://react.dev/>, [16.9.2024.]
- [14] Node.js, dostupno na: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>, [16.9.2024.]
- [15] Visual Studio Code, dostupno na: <https://code.visualstudio.com/docs>, [16.9.2024.]
- [16] Korištenje JSX-a, dostupno na: <https://beta.reactjs.org/learn/writing-markup-with-jsx>, [16.9.2024.]
- [17] Prettier, dostupan na:  
<https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>, [16.9.2024.]

## SAŽETAK

U ovom završnom radu izrađena je web aplikacija za prodaju nekretnina. Web aplikacija omogućuje prikaz i pretraživanje nekretnina na tržištu te dodavanje novih nekretnina na tržište, a napravljena je pomoću HTML-a, CSS-a, MongoDB-a, Express.js-a i React-a. Prikazana je usporedba s već postojećim rješenjima te njihova sličnost s izrađenom aplikacijom. Aplikacija je lagana za korištenje te ju je moguće dodatno proširiti dodavajući mogućnost slanja poruka drugim korisnicima ili plaćanja putem aplikacije.

**Ključne riječi:** korisnik, nekretnina, prodaja, rezervacija, web aplikacija

## **ABSTRACT**

### **Web application for real estate sales**

In this thesis, a web application for real estate sales was developed. The web application allows for the display and search of real estate listings on the market, as well as the addition of new properties to the market. It is built using HTML, CSS, MongoDB, Express.js, and React. A comparison is made with existing solutions, highlighting their similarities with the developed application. The application is easy to use and can be further expanded by adding features such as messaging between users or in-app payments.

**Keywords:** user, real estate, sales, reservation, web application

## **PRILOZI**

Prilog 1. GitLab poveznica na repozitorij aplikacije:

<https://gitlab.com/ivan.koturic/nekretnine-app>