

# Smještaj studenata u stambene jedinice privatnih vlasnika uz pomoć web aplikacije

---

**Buljević, Juraj**

**Master's thesis / Diplomski 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:200:961253>

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

*Download date / Datum preuzimanja:* **2024-12-28**

*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**

**Smještaj studenata u stambene jedinice privatnih vlasnika  
uz pomoć web aplikacije**

**Diplomski rad**

**Juraj Buljević**

**Osijek, godina. 2024**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju****Ocjena diplomskog rada na sveučilišnom diplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Juraj Buljević
<b>Studij, smjer:</b>	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i
<b>Mat. br. pristupnika, god.</b>	D-1361, 07.10.2021.
<b>JMBAG:</b>	0165073023
<b>Mentor:</b>	prof. dr. sc. Dominika Crnjac Milić
<b>Sumentor:</b>	prof. dr. sc. Krešimir Nenadić
<b>Sumentor iz tvrtke:</b>	Nikola Gašparec
<b>Predsjednik Povjerenstva:</b>	izv. prof. dr. sc. Zdravko Krpić
<b>Član Povjerenstva 1:</b>	prof. dr. sc. Dominika Crnjac Milić
<b>Član Povjerenstva 2:</b>	izv. prof. dr. sc. Alfonzo Baumgartner
<b>Naslov diplomskog rada:</b>	Smještaj studenata u stambene jedinice privatnih vlasnika uz pomoć web aplikacije
<b>Znanstvena grana diplomskog rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Broj punoljetnih osoba koje se odlučuje za studiranje i cjeloživotno obrazovanje je sve veći. Često se iz tog razloga javlja potreba za preseljenjem u gradove u kojima su prisutne više i visoko obrazovne ustanove. Studentskih smještaja čiji je vlasnik država nema previše i pojavljuje se nužnost za potragom smještaja u privatnom vlasništvu. Na tržištu postoje različiti mediji koji omogućavaju oglašavanje smještaja no radi povećanja sigurnosti, jednostavnosti i brzine pronalaska odgovarajuće stambene jedinice ovim radom je potrebno dati interaktivnu web aplikaciju. Putem te
<b>Datum ocjene pismenog dijela diplomskog rada od strane mentora:</b>	06.09.2024.
<b>Ocjena pismenog dijela diplomskog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum obrane diplomskog rada:</b>	18.09.2024
<b>Ocjena usmenog dijela diplomskog rada (obrane):</b>	Izvrstan (5)
<b>Ukupna ocjena diplomskog rada:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:</b>	18.09.2024.

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

Osijek, 18.09.2024.

Ime i prezime Pristupnika:	Juraj Buljević
Studij:	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i informatika
Mat. br. Pristupnika, godina upisa:	D-1361, 07.10.2021.
Turnitin podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom: **Smještaj studenata u stambene jedinice privatnih vlasnika uz pomoć web aplikacije**

izrađen pod vodstvom mentora prof. dr. sc. Dominika Crnjac Milić

i sumentora prof. dr. sc. Krešimir Nenadić

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>4</b>
<b>2. PREGLED TRŽIŠTA .....</b>	<b>6</b>
<b>2.1. Analiza preferencija prilikom traženja studentskog smještaja .....</b>	<b>6</b>
<b>2.2. Pregled postojećih rješenja .....</b>	<b>8</b>
2.2.1. Booking platforma .....	8
2.2.2. Hostelworld platforma .....	9
2.2.3. Airbnb platforma .....	10
2.2.4. Njuškalo oglasnik .....	11
2.2.5. Facebook grupe za unajmljivanje smještaja .....	12
<b>3. ANALIZA ZAHTJEVA.....</b>	<b>13</b>
<b>3.1. Identifikacija glavnih funkcionalnosti aplikacije .....</b>	<b>13</b>
<b>3.2. Opis funkcionalnosti .....</b>	<b>14</b>
3.2.1. Registracija novog korisnika .....	14
3.2.2. Pretraživanje smještaja i slanje upita .....	15
3.2.3. Komunikacija s najmodavcem.....	15
<b>4. ODABRANE TEHNOLOGIJE I ALATI ZA RAZVOJ .....</b>	<b>16</b>
<b>4.1. Docker sustav za upravljanje kontejnerima.....</b>	<b>16</b>
<b>4.2. <i>Symfony</i> razvojni okvir.....</b>	<b>16</b>
4.2.1. Doctrine ORM biblioteka za objektno-orijentirani rad.....	17
<b>4.3. NextJS razvojni okvir .....</b>	<b>18</b>
<b>4.4. Pomoćni alati .....</b>	<b>18</b>
4.4.1. Integrirana razvojna okruženja za razvoj aplikacija .....	19
Testiranje aplikacije putem.....	19
4.4.2. Postman alata.....	19
<b>5. RAZVOJ POSLUŽITELJSKOG DIJELA WEB APLIKACIJE.....</b>	<b>20</b>
<b>5.1. Shema baze podataka .....</b>	<b>21</b>

<b>5.2. Model oglasa i pomoćnih entiteta .....</b>	<b>22</b>
5.2.1. Pohrana koordinata .....	24
5.2.2. Pohrana fotografija putem <i>Cloudinary</i> usluge .....	27
<b>5.3. Upravljanje oglasom .....</b>	<b>28</b>
5.3.1. Kreiranje oglasa .....	28
5.3.2. Upravljanje i ažuriranje oglasa .....	30
5.3.3. Upravljanje rezervacijama .....	32
<b>5.4. Pretraživanje smještaja .....</b>	<b>33</b>
5.4.1. Generiranje SQL upita .....	36
5.4.2. Straničenje rezultata pretrage .....	38
<b>6. RAZVOJ KORISNIČKOG SUČELJA.....</b>	<b>39</b>
<b>6.1. Početna stranica .....</b>	<b>39</b>
<b>6.2. Pretraživanje smještaja .....</b>	<b>40</b>
6.2.1. Stranica sa rezultatima pretrage .....	43
6.2.2. Napredno filtriranje rezultata.....	44
<b>6.3. Stranica oglasa .....</b>	<b>45</b>
<b>6.4. Obrazac za izradu oglasa u više koraka.....</b>	<b>46</b>
<b>6.5. Upravljanje oglasima .....</b>	<b>50</b>
<b>6.1. Komunikacijski sustav i stranica s razgovorima.....</b>	<b>52</b>
<b>7. ZAKLJUČAK.....</b>	<b>54</b>
<b>LITERATURA .....</b>	<b>56</b>
<b>SAŽETAK.....</b>	<b>58</b>
<b>ABSTRACT .....</b>	<b>59</b>
<b>PRILOZI.....</b>	<b>60</b>
<b>ŽIVOTOPIS.....</b>	<b>61</b>

# 1. UVOD

Studentska populacija tipično uključuje mlade punoljetne osobe između 18 i 25 godina, te se procjenjuje da je broj studenata u svijetu oko 254 milijuna u 2024. godini [1], od kojih je 18 milijuna u Europskoj Uniji [2].

Akadska središta tipično su ograničena na uska područja pa velik broj studentske populacije dolazi iz drugih krajeva. Stoga, velik broj studenta ima potrebu na vrijeme pronaći adekvatan smještaj.

Određeni dio studentske populacije uživa u vrlo cjenovno pristupačnom smještaju poput državno financiranih studentskih domova. Međutim broj mjesta unutar domova je relativno ograničen u odnosu na broj studenata. Osim toga student mora ispuniti određene uvjete kako bi mogao sudjelovati u natječaju za mjesto u domu. Iz tog razloga dio studentske populacije odlučuje se, ili je primoran, pronaći privatni smještaj.

Postoje brojne web aplikacije koje pomažu prilikom pronalaska i iznajmljivanja smještaja. Međutim prema dosadašnjim saznanjima ne postoji niti jedna web aplikacija isključivo namijenjena za pronalazak studentskog smještaja. Kako je broj studenata relativno velik, te su zahtjevi studenata specifični pojavljuje se potreba za takvom vrstom specijalizirane web aplikacije.

Studentski smještaji razlikuju se od smještaja namijenjenog generalnoj populaciji. Studenti imaju često vrlo specifične želje koje smještaj mora ispunjavati kako bi bio privlačan studentskoj populaciji. Ponajprije, privatni studentski smještaji moraju biti cjenovno pristupačni. Studenti su često financijski ovisni o roditeljima za vrijeme visokoškolskog obrazovanja pa su roditelji ti koji snose većinu troškova njihovog studentskog života u koje ulaze i troškovi smještaja.

Cijena najma nije jedina stavka bitna kod pronalaska privatnog studentskog smještaja. Vrlo bitna stavka je ponajprije udaljenost do obrazovne ustanove ali udaljenost i do ostalih lokacija koje su bitne studentima poput trgovina, studentskih restorana, rekreacijskih centara i tako dalje.

Osim navedenog studentski smještaj je specifičan po pitanju prisutnosti studentske populacije kroz trajanje kalendarske godine. Konkretno, tijekom ljetnog perioda, koji obično traje od srpnja do listopada, veliki dio studenata napušta svoje smještaje zbog završetka akademske godine i povratka kućama. Kako bi se smanjili troškovi jedan dio studentske populacije potpuno otkazuje smještaj što također utječe i na ponudu smještaja, odnosno odluku najmodavca da ponudi smještaj studentima.

Zadatak ovog diplomskog rada bio je razviti interaktivnu web aplikaciju koja će studentima, za vrijeme obrazovanja, omogućiti jednostavniji pronalazak smještaja. Razvijena aplikacija omogućava objavljivanje slika, postavljanje opisa stambenih jedinica, informacija o okruženju i cijeni najma. Studentima je omogućeno i detaljno filtriranje smještaja po kriterijima poput lokacije, cijene najma, veličine smještaja, pogodnostima i ostalo. Aplikacije također pruža i mogućnost stupanja u komunikaciju s najmodavcem putem ugrađenog komunikacijskog sustava. Osim toga za razliku od standardnih platformi za unajmljivanje studenti neće biti u mogućnosti unaprijed rezervirati smještaj već isključivo najmodavci.

Osim navedenog, aplikacija implementira i sustav rezervacija. Nakon uspješnog dogovora sa studentom, najmodavac ima mogućnost označiti da je smještaj zauzet za ugovoreni period najma. Prilikom pretrage aplikacija uzima u obzir rezervacije smještaja te studentima prikazuje samo one oglase koji su dostupni unutar njihovog željenog perioda najma.

Ovim radom, unutar idućih poglavlja, opisan je razvoj web aplikacije namijenjen pronalasku stambenih jedinica privatnih vlasnika za smještaj studenata. Unutar drugog poglavlja ovog rada razmotrena su već postojeća rješenja namijenjena pronalasku smještaja, a osim toga razmotrena su i istraživanja vezana uz preferencije i želje studenta pri traženju smještaja. Nadalje u trećem poglavlju detaljno su analizirane potrebne funkcionalnosti ove web aplikacije te su dani scenariji upotrebe. U četvrtom poglavlju razmotrene su tehnologije i potrebni alati korišteni prilikom razvoja ove web aplikacije. Peto poglavlje bavi se bazom podataka ali razvojem poslužiteljske aplikacije dok je šesto poglavlje usredotočeno na razvoj klijentske aplikacije primjenom „mobile first“ pristupa .



## **2. PREGLED TRŽIŠTA**

Pregled tržišta predstavlja ključan korak u razvoju web aplikacije, jer omogućava razumijevanje trenutne situacije i potreba na tržištu. U nastavku ovog poglavlja, analizirane su preferencije studenata prilikom traženja smještaja te je pružen pregled postojećih web rješenja za pronalazak smještaja. Kroz analizu preferencija i pregled postojećih rješenja mogu se dobiti vrijedni uvidi koji mogu usmjeriti daljnji razvoj aplikacije. Analiza preferencija omogućuje identifikaciju ključnih trendova i zahtjeva korisnika, dok pregled konkurentnih rješenja pruža uvid u trenutno dostupne opcije na tržištu.

### **2.1. Analiza preferencija prilikom traženja studentskog smještaja**

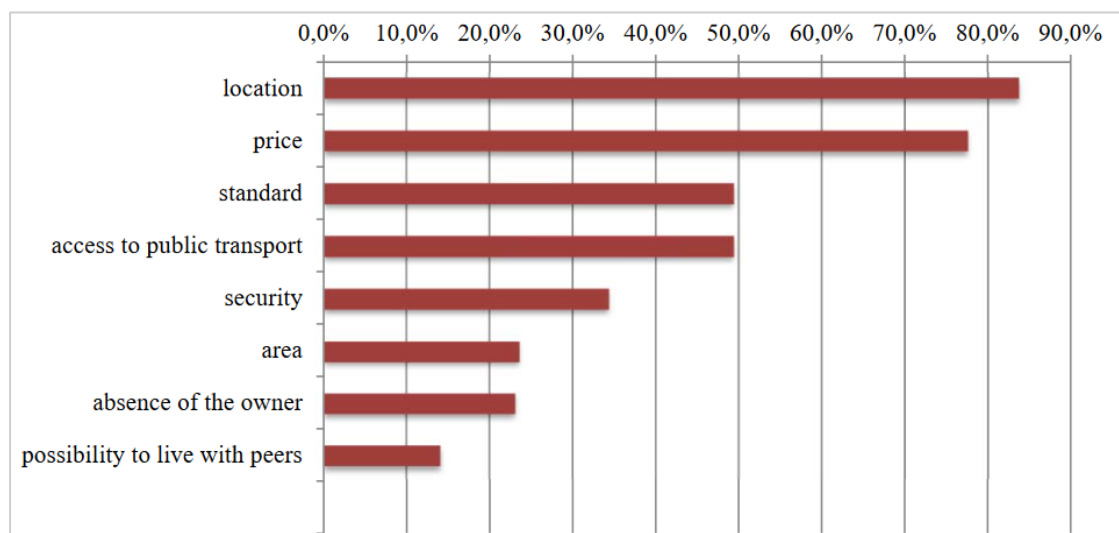
Prema [3] i [4] moguće je vidjeti kako postoji tendencija studenata za pronalazak specifičnog tipa smještaja na određenim lokacijama.

Uzimajući u obzir rezultate ovih analiza, moguće je identificirati osnovni skup funkcionalnosti koje aplikacija treba imati kako bi olakšala proces pronalaska smještaja prema kriterijima koje su istraživanja istaknula kao ključnim.

Osim toga moguće je oblikovati algoritam pretrage koji prilagođava redoslijed pronađenih smještaja, dajući prioritet onima koji najviše odgovaraju optimalnom smještaju utvrđenom analizom preferencija korisnika. Na taj način ostvaruje se znatno kvalitetnija i efikasnija pretraga studentskog smještaja u odnosu na već postojeće web aplikacije koje ne pružaju takav skup funkcionalnosti.

Također, rezultate istraživanja provedenih u radovima [3] i [4] moguće je prenijeti i najmodavcima tijekom procesa kreiranja oglasa kako bi im pomoglo istaknuti određene pogodnosti koje smještaj posjeduje, a za koje možda nisu svjesni da su bitne studentskoj populaciji.

Istraživanja generalno ukazuju na to da su najvažniji kriteriji koji utječu na izbora smještaja: vrsta nekretnine, cijena najma, površina smještaja i udaljenost od obrazovne ustanove.



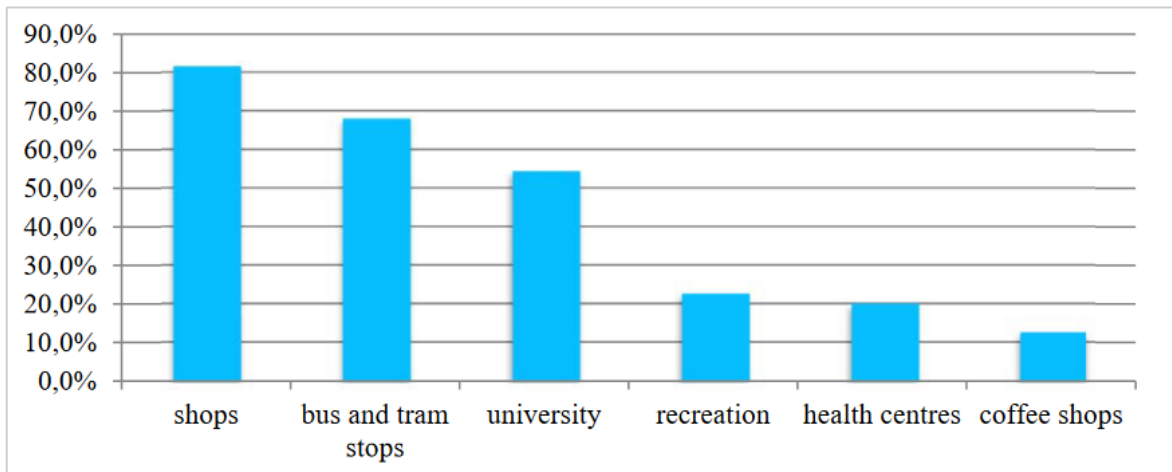
Slika 2.1. Prikaz najvažnijih elemenata koji utječu na izbor smještaja. (redom od najvažnijeg: lokacija, cijena, standard, dostupnost javnog prijevoza, sigurnost, okolina, prisutnost vlasnika, suživot s drugim osobama) [3]

Vrsta nekretnine ima vrlo veliku važnost pri izboru studentskog smještaja. Najpopularniji oblici nekretnina su stanovi, dok su najmanje popularne pojedinačne sobe unutar zajedničkog smještaja. Osim toga vrlo veliku važnost predstavljaju zajedničke prostorije, a najlošije rezultate imaju smještaji gdje se dijeli sanitarni čvor [4].

Cijena najma predstavlja drugi najvažniji kriterij, koji utječe na izbor smještaja, pri čemu porast cijene najma smanjuje zainteresiranost za smještaj.

Udaljenost smještaja do obrazovne ustanove predstavlja idući važan kriterij. Istraživanje provedeno u radu [4] nije pokazalo linearnu ovisnost udaljenosti na izbor smještaja a razlog tome je što velik broj studenata često koristi prijevozna sredstva poput bicikla i javnog prijevoza. Osim toga navedeno istraživanje je utvrdilo kako postoji određeno maksimalno trajanje putovanja, obično od oko 20 minuta, koje predstavlja granicu interesa studenata za smještaj.

Prema radu [3] može se uočiti kako je udaljenost do obrazovne ustanove tek na trećem mjestu po važnosti, od čega su važnije udaljenosti okolnih trgovina te tramvajskih, odnosno autobusnih stanica.



Slika 2.2. Prikaz najvažnijih elemenata koje utječu na izbor lokacije. (redom od najvažnijih do najmanje važnih: trgovine, stanice javnog prijevoza, obrazovna ustanova, rekreacijski objekti, zdravstvene ustanove, kafići).[3]

Slično tome je utvrđeno kako površina smještaja od oko  $12 \text{ m}^2$  (po osobi) predstavlja točku gdje zainteresiranost za smještaj znatno opada.

## 2.2. Pregled postojećih rješenja

Na tržištu postoje brojne aplikacije koje korisnicima pomažu pri pronalaženju i unajmljivanju smještaja. Tipično, ove aplikacije mogu se podijeliti na dvije kategorije: web aplikacije za kratkoročni najam i one za dugoročni najam. Prilikom istraživanja postojećih rješenja, razmotrene su različite funkcionalnosti te specifičnosti načina rada odabranih platformi.

Općenito, uočene razlike između ponuđenih rješenja pretežno su tehničke prirode, dok su osnovne funkcionalnosti vrlo slične. Razlike generalno proizlaze iz ciljane demografske skupine korisnika, ali i ciljane skupine ponuđača smještaja poput hotelijerske industrije ili privatnih vlasnika smještaja. Osim toga ove aplikacije pružaju mogućnost rezervacije smještaja bez kontakta s najmodavcem što u nije primjenjivo u slučaju ove aplikacije.

### 2.2.1. Booking platforma

U segmentu kratkoročnog iznajmljivanja smještaja, Booking [5] predstavlja jednu od dominantnih platformi. Općenito Booking platforma omogućava iznajmljivanje smještaja privatnih vlasnika ali platforma je velikim dijelom usmjerena i na hotelijersku industriju.

Pretraživanje smještaja započinje unosom tri stavke za filtriranje smještaja, a to su lokacije, period najma i broj osoba. Rezultati pretrage su prikazani u obliku popisa, a na stranici sa rezultatima otvaraju se mogućnosti dodatnog filtriranja.

Kako je Booking usmjeren i na hotelijersku industriju platforma pruža funkcionalnosti za upravljanje i unajmljivanje pojedinačnih soba. Na slici 2.3. prikazano je grafičko sučelje koje prikazuje obrazac za izbor sobe.

**Availability**  
Prices converted to EUR

Wed, May 8 — Sun, May 12 | 2 adults - 0 children - 1 room | [Change search](#)

**Filter by:**  
 Rooms  Suites

Room Type	Number of guests	Price for 4 nights	Your Choices	Select amount
<b>Superior Queen or Twin Bed Garden or Pool View</b> Select your bed (if available) <input type="radio"/> 1 queen bed <input type="radio"/> 2 twin beds It features a balcony or terrace overlooking the garden or pool. 25 m <sup>2</sup> Garden view Pool view Air conditioning Minibar Shower Safe TV Telephone Tea/Coffee maker Hairdryer	2	<del>€ 564</del> <b>€ 533</b> Includes taxes and fees 5% off	All-Inclusive Includes parking Free cancellation before May 6, 2024 Pay in advance Genius discount available	0
Deluxe Family Queen or Twin Bed Garden View Select your bed (if available) <input type="radio"/> 2 twin beds <input type="radio"/> 1 full bed and 1 bunk bed Includes a lounge area and offers garden views. 35 m <sup>2</sup> Garden view Air conditioning Terrace Minibar	2	<del>€ 673</del> <b>€ 639</b> Includes taxes and fees 5% off	All-Inclusive Includes parking Free cancellation before May 6, 2024 Pay in advance Genius discount available Free private taxi from the airport to this property	0
Superior Queen or Twin Bed Garden or Pool View Select your bed (if available) <input type="radio"/> 1 queen bed <input type="radio"/> 2 twin beds It features a balcony or terrace overlooking the garden or pool. 25 m <sup>2</sup> Garden view Pool view Air conditioning Minibar Shower Safe TV Telephone Tea/Coffee maker Hairdryer	2	<del>€ 564</del> <b>€ 533</b> Includes taxes and fees 5% off	All-Inclusive Includes parking Free cancellation before May 6, 2024 Pay in advance Genius discount available	0
Deluxe Family Queen or Twin Bed Garden View Select your bed (if available) <input type="radio"/> 2 twin beds <input type="radio"/> 1 full bed and 1 bunk bed Includes a lounge area and offers garden views. 35 m <sup>2</sup> Garden view Air conditioning Terrace Minibar	2	<del>€ 673</del> <b>€ 639</b> Includes taxes and fees 5% off	All-Inclusive Includes parking Free cancellation before May 6, 2024 Pay in advance Genius discount available Free private taxi from the airport to this property	0

[I'll reserve](#)  
• Confirmation is immediate

Slika 2.3. Obrazac za odabir sobe na Booking platformi. Izvor slike [5]

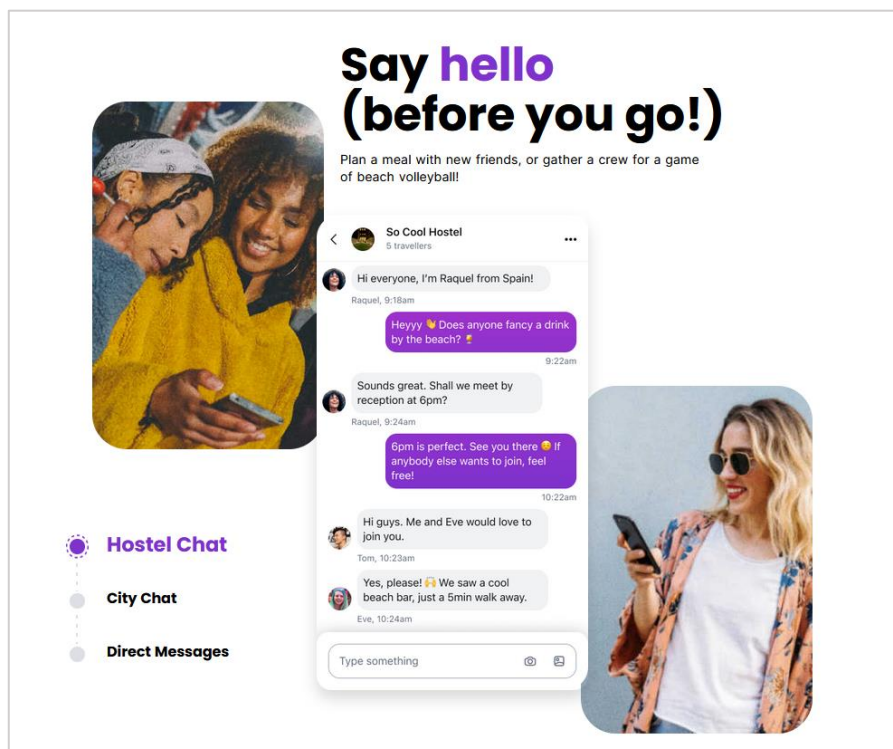
U kontekstu ovog rada, obrazac za izbor sobe je koristan primjer jer postoji ekvivalentna usluga tome segmentu unajmljivanja smještaja studentima u obliku privatnih domova. Stoga bi implementacija sličnog sustava bila korisna jer bi vlasnicima domova omogućila jednostavnije iznajmljivanje putem web aplikacije.

## 2.2.2. Hostelworld platforma

Jedna od specifičnih platformi je *Hostelworld* [6], koja se usredotočuje na hostele. Vrlo zanimljiva funkcionalnost te platforme je omogućavanje gostima uvid u druge goste hotela unutar odabranog perioda gostovanja, te osim toga pruža i funkcionalnost koja omogućava i međusobnu komunikaciju između gostiju putem ugrađene „chat“ aplikacije.

U slučaju iznajmljivanja smještaja studentima, česta pojava je međusobno dijeljenje smještaja između više studenata, pa bi funkcionalnost nalik onoj koju nudi platforma *Hostelworld* bila korisna.

Ako smještaj omogućuje suživot više osoba, vlasnik te nekretnine mogao bi iskoristiti funkcionalnost omogućavanja direktnog komuniciranja između trenutnog stanara i potencijalnog sustanara. Navedena funkcionalnost olakšala bi pronalazak odgovarajućeg i cjenovno pristupačnijeg smještaja, kako trenutnom stanaru tako i potencijalnom sustanaru.



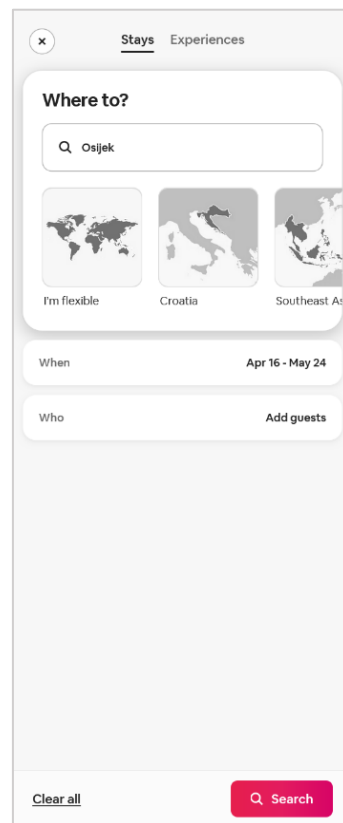
Slika 2.4. Reklamni sadržaj za „chat“ funkcionalnost *Hostworld* platforme. Izvor slike [6]

### 2.2.3. Airbnb platforma

*Airbnb* [7] je platforma koja se tipično koristi za kratkoročni najam smještaja te je usredotočena na pružanje smještaja turistima i drugim putnicima, pri čemu smještaj nude privatni vlasnici.

Općenito najvažnija funkcionalnost ovih platformi je pretraga smještaja, a *Airbnb* platforma nudi vrlo kvalitetnu i detaljnu pretragu smještaja. Pruža niz opcija filtriranja poput filtriranja po lokaciji, cijeni noćenja i različitim pogodnostima. Korisnicima se također omogućava i pretraživanje po vrsti nekretnina i vrsti smještaja.

Samo pretraživanje se provodi u dva koraka: korisnik prvo odabire lokaciju i željeni period najma, nakon čega mu se prikazuju grubi rezultati pretrage. Zatim, na stranici s rezultatima, korisnik može dodatno filtrirati smještaj prema preciznijim kriterijima. Na slici 2.5. prikazan je izgled komponente za filtriranje smještaja.



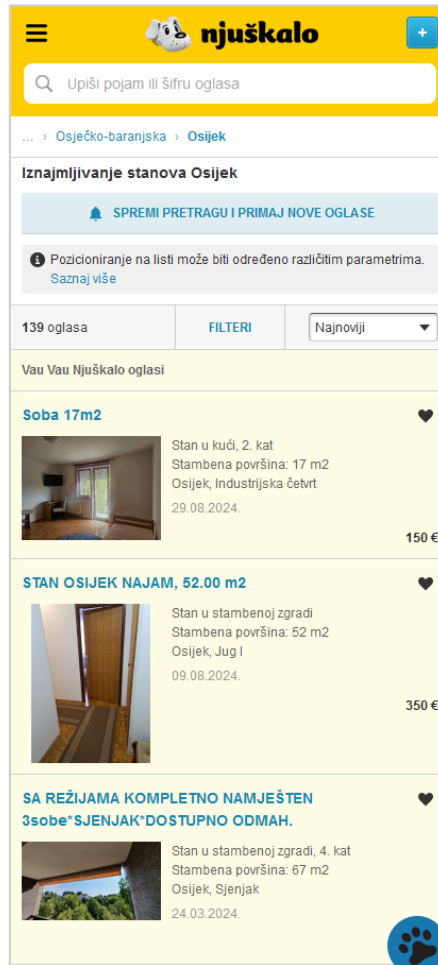
Slika 2.5. Obrazac za pretragu – izbor lokacije. Izvor slike [7]

Rezultati pretrage prikazuju se u obliku popisa, a svaki element popisa prikazuje pojedini smještaj s osnovnim informacijama poput cijene najma, lokacije, ocijene korisnika, ali i fotografije smještaja.

#### 2.2.4. Njuškalo oglasnik

Njuškalo je jedan od najpopularniji oglasnika na području Republike Hrvatske. Omogućava privatnim osobama reklamiranje usluga, proizvoda ali i nekretnina. Pretraživanje nekretnina za potrebe najma vrši se putem ugrađenih kategorija ili putem tekstualne pretrage.

Odabirom na kategoriju iznajmljivanja, korisniku se otvara popis nekretnina. Na slici 2.6. prikazana je stranica sa popisom rezultata pretrage, a dodatno na stranici se nalaze brojne postavke za pretragu. Korisnik može pretraživati nekretnine po lokaciji, cijeni najma, površini prostora i drugo.



Slika 2.6. Njuškalo – Stranica sa popisom rezultata pretrage. Izvor slike [8]

Komunikacija sa najmodavcem moguća je putem ugrađenog komunikacijskog sustava ili direktnim kontaktom putem telefonskog broja najmodavca.

### 2.2.5. Facebook grupe za unajmljivanje smještaja

Jedan od čestih načina pronalaska nekretnina za dugoročni najam je putem Facebook web aplikacije. Konkretno Facebook omogućava korisnicima kreiranje specijaliziranih grupa, pa stoga postoje brojne grupe koje su usredotočene unajmljivanju nekretnina. Tipično ove grupe su usredotočene na uža područja poput određenog grada što olakšava korisnicima da pronađu relevantne ponude.

Osim navedenog ne postoji dobar način za detaljnije pretraživanje oglasa. Informacije o smještaju generalno su navedene tekstu objave, a pri tom su često priložene i fotografije smještaja.

### 3. ANALIZA ZAHTJEVA

Analiza zahtjeva predstavlja ključnu fazu pri razvoju web aplikacije. Omogućuje jasno definiranje potrebnih funkcionalnosti i specifičnih karakteristika koje aplikacija mora imati. Precizno definirani zahtjevi osiguravaju da postoji jasan smjer prilikom razvoja. U idućim poglavljima identificirane su glavne funkcionalnosti aplikacije, kao i scenariji upotrebe koji pružaju uvid u način korištenja aplikacije iz pogleda krajnjih korisnika.

#### 3.1. Identifikacija glavnih funkcionalnosti aplikacije

U ovom poglavlju fokus je na identifikaciji glavnih funkcionalnosti web aplikacije. Cilj je razumjeti ključne zahtjeve i potrebe korisnika, te identificirati funkcionalnosti koje aplikacija mora pružiti korisnicima.

- **Registracija korisnika**
  - Korisnici trebaju imati mogućnost stvaranja računa, što je neizbježno ukoliko žele ostvariti pristup svim funkcionalnostima web aplikacije.
  - Registracija treba uključivati unos osnovnih informacija kao što su: korisničko ime, adresa e-pošte te zaporka.
  - Prilikom registracije korisnik treba unijeti i određene privatne informacije poput imena i prezimena te odabrati vrstu profila (najmodavac ili najmoprimac).
- **Kreiranje oglasa**
  - Korisnici s ulogom najmodavca trebaju imati mogućnost kreiranja i objavljivanja oglasa. Kreiranje oglasa zahtijeva unos svih potrebnih informacija o smještaju i postavljanje fotografija.
- **Pretraživanje smještaja**
  - Najmoprimci trebaju imati mogućnost pregledavanja dostupnih smještaja putem aplikacije.
  - Ova funkcionalnost treba omogućiti filtriranje i pretraživanje smještaja prema kriterijima poput lokacije, cijene najma, željenog perioda najma te prema traženim pogodnostima.
  - Prikazani rezultati pretraživanja trebaju biti pregledni i informativni.
- **Pregled detalja o smještaju**
  - Najmoprimci također moraju imati pristup detaljnim informacijama o svakom pojedinom smještaju.



- Detalji o smještaju trebaju uključivati informacije poput opisa smještaja koji može sadržavati informacije o broju prostorija, opremljenost smještaja te ostalim informacijama usko vezanim uz smještaj.
- Osim toga najmoprimcima treba omogućiti prikaz fotografija smještaja te prikaz lokacije putem informativne karte.
- Cilj je najmoprimcima pružiti sve potrebne informacije kako bi mogli donijeti informiranu odluku o izboru smještaja.
- **Slanje upita i komunikacija s najmodavcem**
  - Najmoprimci trebaju imati mogućnost slanja upita te izravne komunikacije s najmodavcem putem aplikacije.
  - Slanje upita predstavlja funkcionalnost kojom najmoprimac započinje razgovor s najmodavcem.
  - Cilj ove funkcionalnosti je pružiti najmoprimcima priliku za postavljanje pitanja, traženje dodatnih informacija, pregovaranje o uvjetima i dogovaranje rezervacije.
- **Rezervacija smještaja**
  - U slučaju uspješnog dogovora najmodavac treba imati mogućnost potvrde rezervacije kojom se navedeni smještaj zaključava za dogovoreni period najma.

## 3.2. Opis funkcionalnosti

U ovom poglavlju fokus je na prikazu tipičnih scenarija upotrebe aplikacije. Cilj je opisati tipične situacije i korisničke interakcije s aplikacijom, te prikazati način na koji korisnici mogu iskoristiti funkcionalnosti aplikacije za pronalazak smještaja.

### 3.2.1. Registracija novog korisnika

Ukoliko korisnik prilikom pristupa izrađenoj web aplikacija želi pristupiti svim funkcionalnostima dužan je najprije kreirati korisnički račun. Korisnik pristupa početnoj stranici aplikacije te odabire opciju „Sign up“. Na stranici za registraciju korisnika prikazan je obrazac za kreiranje korisničkog računa. Putem obrasca korisnik unosi potrebne informacije poput adrese e-pošte i zaporke te unosi i privatne informacije poput imena i prezimena. Konačno korisnik odabire vrstu računa (najmodavac ili najmoprimac). Nakon uspješnog unosa svih potrebnih podataka korisnik odabire opciju „Register“. Ukoliko su svi podaci uneseni ispravno aplikacija obavještava korisnika da je profil kreiran. Nakon toga korisnik ima pristup svim funkcionalnostima odabrane uloge.

### **3.2.2. Pretraživanje smještaja i slanje upita**

Pretraživanje smještaja je dostupno svim posjetiteljima web aplikacije, neovisno jesu li prijavljeni ili ne. Međutim ukoliko najmoprimac želi pristupiti funkcionalnosti slanja upita mora se najprije prijaviti. Nakon uspješne prijave, najmoprimac je preusmjeren na početnu stranicu koja ujedno služi kao i stranica za pretraživanje smještaja.

Kao dio funkcionalnosti pretraživača najmoprimac može postaviti različite kriterije pretrage kao što su lokacija, cijena, period iznajmljivanja i skup željenih pogodnosti. Nakon postavljanja kriterija najmoprimac odabire opciju „Search“ kako bi dobio rezultate pretrage. Ukoliko postoje smještaji koji zadovoljavaju odabrane kriterije, rezultati se prikazuju najmoprimcu. Najmoprimac pregledava rezultate i odabire željeni smještaj.

Nakon pregleda detalja, najmoprimac odabire opciju „Send Inquiry“ i unosi detalje upita poput željenog perioda boravka. Aplikacija provjerava dostupnost smještaja za odabrani period, te ako je dostupan, sustav evidentira upit a najmoprimac dobiva potvrdu da je upit uspješno poslan dok najmodavac dobiva obavijest o zaprimljenom upitu. Najmoprimac nakon toga može stupiti u komunikaciju s najmodavcem.

### **3.2.3. Komunikacija s najmodavcem**

Web aplikaciji nudi mogućnost stupanja u kontakt s najmodavce, što omogućava postavljanje pitanja radi pristupa dodatnim informacijama o smještaju. Odabirom željenog smještaja najmoprimca se preusmjerava na stranicu s detaljima smještaja gdje se prikazuje obrazac za slanje upita. Najmoprimac ispunjava sve potrebne podatke te upisuje poruku u za to predviđenu lokaciju, nakon čega odabire opciju „Send Inquiry“.

Aplikacija evidentira upit, obavještava najmoprimca o uspješnom slanju upita te obavještava najmodavca o novom upitu. Aplikacija omogućava praćenje obavijesti o odgovorima putem klijentske aplikacije.

Najmoprimac može u bilo kojem trenutku pristupiti popisu svih upita odabirom opcije „Messenger“. Odabirom na razgovor iz popisa prijašnjih razgovora, najmoprimcu se prikazuje stranica za komunikaciju koja sadrži svu prijašnju komunikaciju s najmodavcem.

## **4. ODABRANE TEHNOLOGIJE I ALATI ZA RAZVOJ**

U ovom poglavlju opisane su ključne tehnologije i alati koji su korišteni prilikom razvoja klijentske i poslužiteljske aplikacije, te dodatni alati koji su bili korisni prilikom razvoja i testiranja aplikacije.

### **4.1. Docker sustav za upravljanje kontejnerima**

Docker [9] je platforma koja omogućava korisnicima kreiranje i pokretanje tzv. kontejnera. Za potrebe aplikacije korišten je kontejner koji pokreće izolirano okruženje gdje se izvršava program poslužiteljskog dijela aplikacije.

Kontejner se temelji na Alpine Linux verziji operativnog sustava, koja predstavlja minimalnu verziju Linux sustava što omogućava optimizaciju performansi i veličine kontejnera, odnosno omogućava da se instaliraju samo one komponente sustava koje su potrebne za ispravan rad.

Konfiguracija kontejnera uključuje podešavanje upravljačkog korisničkog profila, podešavanje direktorija (poput „www“ foldera) te instalacije svih potrebnih PHP ekstenzija. Također instalirana je Composer biblioteka koja olakšava rad sa Symfony razvojnim okvirom, kao i xdebug ekstenzija koja olakšava otkrivanje i otklanjanje pogrešaka.

Osim navedenog kontejnera, korištena je i Docker slika kontejnera optimiziranog za rad MySQL baze podataka.

Ovim putem dobiveno je izolirano i konstantno okruženje koje eliminira probleme koji mogu nastati zbog različitosti u postavkama i verzijama između korištenih biblioteka, programa i ostalih elemenata sustava.

### **4.2. *Symfony* razvojni okvir**

Razvojni okviri (engl. frameworks) predstavljaju skup alata i programa za razvoj računalnih aplikacija. Glavna prednost je da pružaju skupove funkcija, klasa i metoda koje se mogu koristiti za ubrzani razvoj aplikacija.

Symfony [10] predstavlja jedan od razvojnih okvira koji je usredotočen na razvijanje web aplikacija. U kontekstu ove aplikacije, Symfony razvojni okvir korišten je za razvoj poslužiteljskog dijela aplikacije. Konkretno poslužiteljska aplikacija je razvijena u obliku REST (engl. Representational State Transfer) web aplikacijskog sučelja.

Pomoću Symfony usmjerivača razvijene su sve potrebne rute za izvršavanje operacija nad podatkovnim resursima entitetima web aplikacije. Operacije se svode na kreiranje, ažuriranje, brisanje i dohvaćanje podataka iz baze.

Osim usmjerivača korišten je i Symfony serijalizator za pretvorbu podataka iz objektnog formata u JSON (*engl.* JavaScript Object Notation) zapis i obratno. Također prilikom kreiranja i uređivanja nekog resursa korišten je Symfony validator za provjeru ispravnosti unesenih podataka.

Također vrlo bitna korištena komponenta Symfony sustava je Symfony vatrozid putem kojeg je implementiran sigurnosni aspekt sustava poput kontrole pristupa rutama.

#### **4.2.1. Doctrine ORM biblioteka za objektno-orijentirani rad**

Kako bi se pojednostavio rad s bazom podataka često se koriste ORM (*engl.* *Object Relational Mapping*) biblioteke koje omogućavaju objektno-orijentirani rad s podacima baze podataka. Za potrebe ove aplikacije korištena je Doctrine ORM biblioteka [12].

Za ispravan rad s Doctrine ORM-om potrebno je entitete prikazane u ER dijagramu pretvoriti u PHP objekte.

Za ispravnu pretvorbu u PHP objekt najvažnija stavka je izabrati odgovarajuće tipove atributa (varijabli) u PHP-u koji će odgovarati tipovima podataka u bazi. Na primjer, ako je u bazi određeni tip podataka „integer“, odgovarajući tip varijable u PHP-u je tada „int“. Slično tome za tekstualna polja u bazi podataka koristi se „string“ tip podatka.

Uz to, Doctrine pruža korisnicima i posebne ugrađene tipove podataka poput *DateTime* klase koja je namijenjena za rad s vremenskim podacima. Također vrlo važan ugrađeni tip je *Collection* koji omogućava rad s poljima.

Osim toga Doctrine pruža i mogućnost korisnički definiranih tipova podataka. Na primjer, u kontekstu ove aplikacije potrebno je bilo razviti tipove podataka za različite *enum* strukture, ali potrebno je bilo razviti i poseban tip podataka za rad s prostornim podacima.

Također za ispravnu pretvorbu vrlo važno je i provesti mapiranje između atributa modela i atributa tablice baze podataka, ali i mapirati relacije između različitih tablica. Doctrine pruža nekoliko metoda mapiranja, a odabrano je mapiranje putem XML datoteke.

### 4.3. NextJS razvojni okvir

Za razvoj korisničkog sučelja korišten je NextJS razvojni okvir. NextJS se temelji na *React* [11] biblioteci otvorenog koda, odnosno proširuje React biblioteku sa brojnim funkcionalnostima koje ubrzavaju postupak razvoja web aplikacije.

Jedna od korištenih NextJS funkcionalnosti koja je vrlo korisna je usmjerivač baziran na mapama, gdje svaka mapa sadrži komponentu s nazivom „page“ koja realizira stranicu i gdje naziv mape predstavlja naziv rute. Na taj način su razvijene sve rute klijentske aplikacije, također na sličan način se realiziraju i API (*engl. application program interface*) rute što je bilo korisno za razvoj funkcionalnosti vezanih uz autentikaciju.

Osim API ruta NextJS ima mogućnost definiranja funkcionalnosti koje se izvršavaju na poslužiteljskoj strani klijentske aplikacije (*engl. Server Actions*). Navedena funkcionalnost korištena je prilikom razvoja obrazaca, konkretno za izvršavanje radnji predaje obrasca, dohvaćanje opcija, te validacije podataka.

Također kroz primjenu funkcionalnosti koje se izvršavaju na poslužitelju, moguće je kontrolirati način na koji se stranica renderira. NextJS podržava kombinaciju poslužiteljskog i statičnog renderiranja stranice. Tim putem je većina stranica web aplikacije je razvijena tako da radi kroz statično renderiranje, a određene stranice poput stranice s rezultatima pretrage se dinamički generiraju.

NextJS također omogućava i druge razne optimizacije, a jedna od važnijih funkcionalnosti je optimizacija renderiranja fotografija. Konkretno NextJS pruža „*next/image*“ komponentu putem koje je realizirana asinkrona dostava fotografija.

Općenito sve navedene funkcionalnosti moguće je ostvariti kroz React korištenjem dodatnih biblioteka, međutim Next.js je znatno kompaktnije i sveobuhvatnije okruženje. Iz tog razloga izabran je za razvoj ove aplikacije.

### 4.4. Pomoćni alati

Prilikom razvoja aplikacije vrlo su korisni određeni pomoćni alati koji olakšavaju proces razvoja web aplikacije. U nastavku opisani su najbitniji pomoćni alati koji su korišteni prilikom razvoja ove web aplikacije.

#### **4.4.1. Integrirana razvojna okruženja za razvoj aplikacija**

Jedan od najvažnijih pomoćnih alata za razvoj aplikacije su integrirana razvojna okruženja, odnosno IDE (*engl. Integrate Development Environment*). Za potrebe razvoja ove aplikacije korištena su dva integrirana razvojna okruženja: PhpStorm za razvoj poslužiteljskog dijela i WebStorm za razvoj klijentskog dijela aplikacije. Razlog izbora ovih integriranih razvojnih okruženja su što imaju vrlo dobru integraciju sa s Symfony, odnosno NextJS razvojnim okruženjima.

#### **4.4.2. Testiranje aplikacije putem Postman alata**

Kako bi se olakšao proces razvoja i testiranja rada web aplikacije korišten je Postman [13] alat. U kontekstu ove aplikacije Postman alat je korišten prilikom razvoja poslužiteljske aplikacije.

Za svaku rutu koju implementira poslužitelj kreiran je HTTP (*engl. hypertext transfer protocol*) zahtjev tako da su definirana potrebna zaglavlja, parametri i tijelo zahtjeva koje poslužitelj očekuje.

Nakon slanja zahtjeva, Postman alat prikazuje odgovor poslužitelja, što olakšava provjeru ispravnosti rada i otklanjanje pogrešaka u radu poslužitelja.

## 5. RAZVOJ POSLUŽITELJSKOG DIJELA WEB APLIKACIJE

Moderne web aplikacije često implementiraju slojevitú arhitekturu koja se tipično sastoji od idućih slojeva:

- Prezentacijski sloj odgovoran je za prikazivanje podataka i interakciju s korisnicima. Ovaj sloj se tradicionalno implementira u sklopu poslužiteljske aplikacije, međutim često se implementira kao i zasebna klijentska aplikacija. Tipično se koriste HTML (*engl. Hypertext Markup Language*) i CSS (*engl. Cascading Style Sheets*) tehnologije, te skriptni jezik JavaScript za razvoj korisničkog sučelja. Osim navedenih tehnologija postoje brojne biblioteke i razvojna okruženja koja ubrzavaju proces razvoja od kojih su najpoznatiji *React* i *Angular*.
- Poslovni sloj implementira poslovnu logiku kojom se ostvaruju funkcionalnosti web aplikacije. Ovaj sloj, tipično se bavi obradom podataka i korisničkih zahtjeva. Osigurava da aplikacija radi ispravno, da su uneseni podaci valjani te da korisnik ima pristup svim potrebnim podacima. Osim navedenog brine se i o sigurnosti aplikacije, odnosno osigurava da su korisnici autentificirani i da su autorizirani za izvršavanje željenih radnji.
- Podatkovni sloj uglavnom je odgovoran za pohranjivanje i dohvaćanje podataka. Najčešće se koriste SQL ili NoSQL baze podataka za pohranjivanje ali mogu se koristiti i vanjski servisi za pohranu podataka.

Integracija između slojeva ključna je za uspješno i skladno funkcioniranje web aplikacije. Svaki sloj ima svoju specifičnu ulogu i odgovornosti. Međusobno su povezani kako bi zajedno pružili potpuno iskustvo korisnicima te omogućili učinkovitu i sigurnu obradu podataka.

Poslužiteljski dio web aplikacije razvijen je u obliku REST (*engl. Representational State Transfer*) API servisa pri čemu je korišteno Symfony razvojno okruženje. Ova komponenta sustava predstavlja jezgru web aplikacije, koja implementira cjelokupnu poslovnu logiku vezanu uz domenu iznajmljivanja smještaja. Osim toga poslužiteljski dio aplikacije upravlja i sigurnosnim aspektima web aplikacije, poput autorizacije i kontrole pristupa podacima.

REST API predstavlja skup principa i ograničenja koja se primjenjuju pri razvoju programskog sučelja. REST pristup je prvobitno opisan u radu [18], ali se tijekom vremena razvio i danas ne postoji samo jedan ispravan način implementacije a u praksi se gotovo nikada ne implementira „čista“ REST arhitektura.

Glavni elementi REST API arhitekture nazivaju se „resursi“ koji se mogu objasniti kao jedinstvene podatkovne jedinice nad kojima aplikacija vrši operacije. U kontekstu ove aplikacije, resurs može predstavljati oglas, rezervaciju ili korisnički profil. Svaki resurs ima jedinstvenu pod-adresu, ili URI (*engl.* Uniform Resource Identifier) na primjer „/api/oglasi“, putem kojeg se vrše operacije poput dohvaćanja, kreiranja ili izmjene podataka.

Kako bi se operacije međusobno razlikovale koriste se različiti HTTP zahtjevi:

- „GET“ za dohvaćanje,
- „POST“ za kreiranje,
- „PUT“ za potpunu izmjenu,
- „PATCH“ za parcijalnu izmjenu,
- „DELETE“ za brisanje resursa.

Jedno od važnih ograničenja je da REST ne čuva stanje klijenta između zahtjeva. Iz tog razloga svaki zahtjev od klijenta prema serveru mora sadržavati sve potrebne informacije za obradu tog zahtjeva.

## 5.1. Shema baze podataka

Osnovni skup podataka s kojim rukuje aplikacija odnosi se na skup podataka koji opisuju nekretninu, oglas te razdoblje najma.

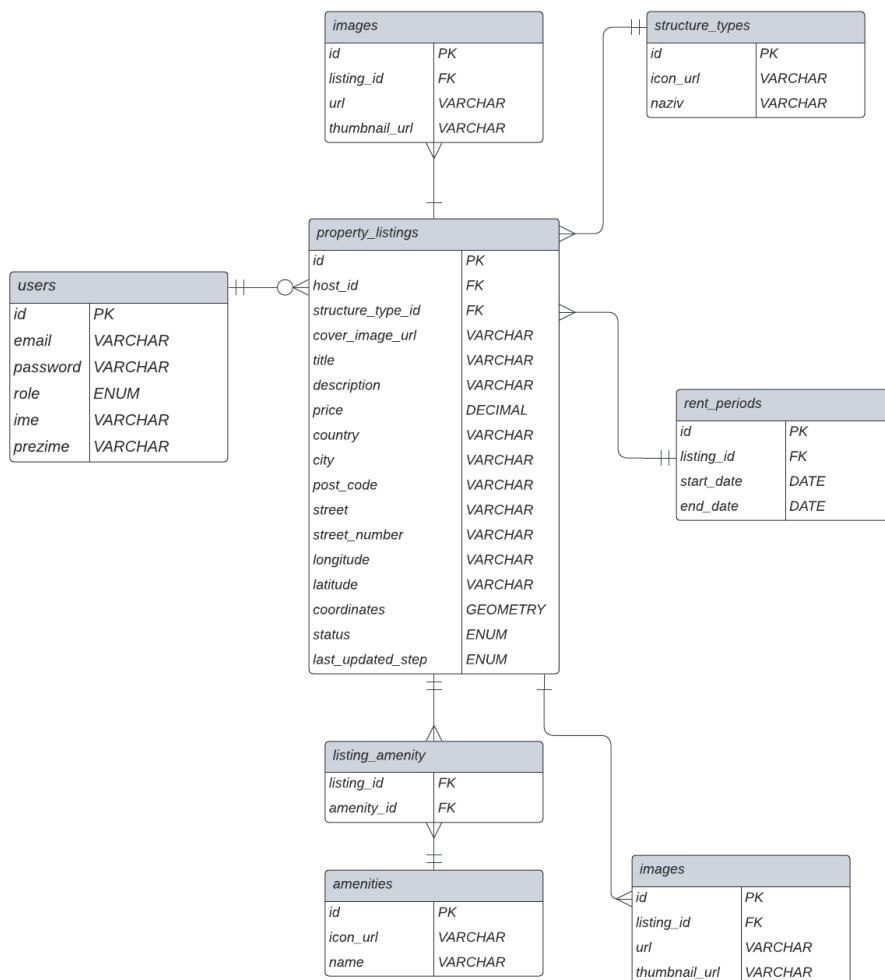
Oglas (*listing*) predstavlja glavni entitet baze podataka. Oglas posjeduje brojne attribute poput naziva, opisa, lokacije i cijene. Osim navedenog, s entitetom oglasa povezano je nekoliko pomoćnih entiteta poput entiteta pogodnosti i entiteta vrste nekretnine. Osim navedenog, uz nekretnine mogu biti povezane i brojne slike, stoga je razvijen i pomoćni entitet sa specifičnom strukturom koja omogućuje njihovo spremanje i manipuliranje.

Također od velike važnosti je i entitet korisnika (*user*), koji omogućava uspostavu „vlasništva“ nad određenim entitetima, osim toga omogućava i kontrolu pristupa nad podacima.

Vrlo važna funkcionalnost web aplikacije je upravljanje rezervacijama, posebice iz razloga što je kako bi se ostvarila funkcionalnost pretrage nekretnina po dostupnosti unutar nekog određenog vremenskog perioda. U tu svrhu razvijen je entitet rezervacije (*rent\_period*), koji sadrži attribute koji definiraju početak i kraj perioda najma, te dodatno i atribut koji veže taj period najma s određenom nekretninom (*listing\_id*).

U nastavku na slici 5.1. prikazana je ER (*engl. Entity Relationship*) dijagram baze podataka.





Slika 5.1. Dijagram baze podataka.

## 5.2. Model oglasa i pomoćnih entiteta

Model oglasa predstavlja glavni model ove web aplikacije. U tablici 5.1. prikazani su nazivi atributa uz njihove odgovarajuće tipove podataka. Većina atributa oslanja se na ugrađene tipove podataka, a preostali atributi poput *host* i *structureType* modeliraju relacije pa stoga koriste njihove modele kao tipove podataka.

Tablica 5.1. Atributi modela oglasa.

Naziv atributa	PHP tip podatka
id	int
host	User
title	string
description	string
price	int
coverImageUrl	string
location	Location
structureType	StructureType
images	Collection
amenities	Collection
status	enum
lastUpdatedStep	enum

Jedna od iznimki je tip podataka definiran klasom *Location*, koji predstavlja takozvanu ugrađenu (*engl.* Embedded) klasu čiji su atributi prikazani u tablici 5.2.

Korištenjem ugrađenih klasa omogućava da se skup međusobno usko povezanih atributa izdvoji u zasebnu klasu ali da se pri tome tvori posebna tablica u bazi podataka. Na taj način model oglasa se pojednostavljuje, ali otvara se i mogućnost ponovne upotrebe tih klasa u kontekstu nekih drugih modela.

Tablica 5.2. Atributi „Embedded“ klase *Location*.

Naziv atributa	PHP tip podatka
country	string
city	string
postCode	string
street	string
streetNumber	string
coordinates	Point

Atribut *host* odgovara modelu korisnika (najmodavca) koji povezuje oglas s najmodavcem te omogućava autorizaciju i kontrolu prava pristupa.

Atribut *amenities* predstavlja niz pogodnosti koje svaka nekretnina pruža. Iz razloga što se radi o nizu relacija odgovarajući tip podatka je *Collection*. Konkretno atribut *amenities* predstavlja polje relacija, odnosno modela tipa *Amenitiy* čiji su atributi dani u tablici 5.3.

U tablici 5.3. dani su i atributi *StructureType* modela koji dijeli iste attribute kao i *Amenitiy* te modelira tip nekretnine.

Tablica 5.3. Atributi modela *StructureType* i *Amentiy*.

Naziv atributa	PHP tip podatka
id	string
name	string
iconUrl	string

Na sličan način atribut *images* predstavlja polje *Image* modela čiji su atributi dani u tablici 5.4. Upravljanje slikama opisano je u poglavlju 5.2.2.

Tablica 5.4. Atributi modela *Image*.

Naziv atributa	PHP tip podatka
id	string
url	string
thumbnailUrl	string
listing	Listing

### 5.2.1. Pohrana koordinata

Prilikom spremanja podataka u bazu, od posebne je važnosti način na koji se spremaju koordinate. U MySQL bazi podataka, postoji mogućnost korištenja posebnih prostornih tipova podataka, što je detaljno objašnjeno u dokumentaciji [16]. Kako bi se precizno zabilježila lokacija smještaja, korišten je „GEOMETRY“ tip podatka.

Konkretno, koordinate se koriste u klasi *Location* čiji su atributi dani u tablici 5.2. Za ispravan rad Doctrine ORM biblioteke s prostornim tipovima razvijene su dvije dodatne klase.

Prva klasa je PHP objekt koji predstavlja *Point* tip podataka, čiji je kod dan na slici 5.2., dok je *GeometryType* druga klasa pomoću koje Doctrine biblioteka vrši konverziju između prostornih podataka unutar baze podataka te objekta klase *Point*.

```
class Point
{
    private float $latitude;
    private float $longitude;

    public function __construct(float $latitude, float $longitude)
    {
        $this->latitude = $latitude;
        $this->longitude = $longitude;
    }

    public function getLatitude(): float
    {
        return $this->latitude;
    }

    public function setLatitude(float $latitude): void
    {
        $this->latitude = $latitude;
    }

    public function getLongitude(): float
    {
        return $this->longitude;
    }

    public function setLongitude(float $longitude): void
    {
        $this->longitude = $longitude;
    }

    public function __toString(): string
    {
        return sprintf('POINT(%f %f)', $this->longitude, $this->latitude);
    }
}
```

Slika 5.2. Klasa *Point*

*GeometryType* klasa sadrži dvije ključne funkcije čiji je kod dan na slici 5.3.

Prva funkcija pod nazivom *convertToPHPValue* vrši konverziju u PHP objekt, dok funkcija *convertToDatabaseValue* konvertira objekt klase *Point* u oblik prikladan za pohranu u bazu podataka.

```

public function convertToPHPValue($value, AbstractPlatform $platform): Point
{
    list($longitude, $latitude) = sscanf($value, 'POINT(%f %f)');

    return new Point($latitude, $longitude);
}

public function convertToDatabaseValue($value, AbstractPlatform $platform)
{
    if (!$value instanceof Point) {
        throw new \Exception('value is not of type Point');
    }

    return sprintf('POINT(%f %f)', $value->getLatitude(), $value->getLongitude());
}

```

Slika 5.3. Funkcije za konverziju: *convertToPHPValue* i *convertToDatabaseValue*.

Prilikom označavanja stupca za spremanje prostornih tipova podataka, potrebno je specificirati i koordinatni sustav koji će se koristiti. Tipično koordinatni sustavi namijenjeni za rad s točkama na zemaljskoj površini koriste geografsku duljinu i širinu. MySQL baza podataka omogućava korištenje različitih koordinatnih sustava, a za potrebe ove aplikacije odabran je WGS84 standard [17] koji odgovara koordinatnom sustavu zemlje.

U kontekstu baze podataka, odabir koordinatnog sustava postiže se putem SRID (*engl. Spatial Reference Identifier*) vrijednosti koja se koristi prilikom definiranja stupca tablice. Korištenjem SRID-a 4326 odgovara upravo WGS84 standardu, omogućava se usklađenost s globalnim normama i precizno lociranje mjesta. Osim toga omogućava se i direktnu usklađenost s *Mapbox* uslugom za geolokaciju i pretraživanje lokacija koja je korištena na klijentskoj strani ove web aplikacije.

Nadalje, korištenjem ovih posebnih tipova podataka omogućava se i pristup ugrađenim funkcijama koje su posebno optimizirane za rad s prostornim podacima. Osim toga, otvara se i mogućnost korištenja prostornih indeksa što značajno ubrzava obradu SQL zahtjeva koji uključuju prostorne podatke. Prostorni indeksi omogućuju bazi podataka efikasnije pretraživanje, filtriranje i izračunava udaljenosti između točaka, što je ključno za aplikacije koje se oslanjaju na geolokaciju ili analizu prostornih podataka.

### 5.2.2. Pohrana fotografija putem *Cloudinary* usluge

Za potrebe upravljanja multimedijским sadržajem korištena je *Cloudinary* CDN (*engl. Content Delivery Network*) usluga, koja omogućava upravljanje, manipuliranje, spremanje i optimizaciju multimedijskog sadržaja za prijenos putem Interneta.

*Cloudinary* nudi brojne mogućnosti za rad s multimedijским sadržajem, od koji su najvažnije funkcionalnosti spremanja, obrade i prijenosa sadržaja. Kako bi se pojednostavilo korištenje ovih usluga korištena je *Cloudinary* SDK (*Software Development Kit*) biblioteka.

Nakon što poslužiteljska aplikacija zaprimi zahtjev za pohranjivanje fotografije, zaprimljena fotografija prosljeđuje se *Cloudinary* sustavu, gdje se potom obrađuje i sprema u lokalnu bazu podataka a poslužitelju se vraća odgovor koji sadrži sve važne informacije za daljnje upravljanje i rad s fotografijom.

Od ključnih informacija, najvažnije su identifikator fotografije, te adresa putem koje se omogućava dohvaćanje pohranjene fotografije. Prilikom slanja zahtjeva za obradu fotografije, također je moguće izvršiti i određene skupove transformacija nad predanom fotografijom, čime se fotografija prilagođava za efikasan prijenos.

Za potrebe ove aplikacije, izvršavaju se dva skupa transformacija koje generiraju dvije prilagođene kopije, dok se original ne koristi.

Prvi skup transformacija smanjuje rezoluciju te siječe fotografiju na kvadratni oblik. Rezultat je kopija fotografije s vrlo malom veličinom, koja je pogodna za korištenje u popisu smještaja te u pozicijama na stranici gdje su fotografije male fizičke veličine.

Drugi skup transformacija održava dimenzije fotografije i izvršava kompresiju. Rezultat ove transformacije je kopija manje veličine koja je pogodna za korištenje kod prikazivanja uvećanih fotografija (na primjer preko cijelog zaslona).

Za spremanje podataka o fotografijama u bazu podataka razvijena je klasa *Image* koja predstavlja model fotografije oglasa, a čiji su atributi prikazani u tablici 5.4.

Model fotografije povezan je s oglasom kroz atribut *listing* i sadrži identifikator fotografije *id* generiran kojeg generira *Cloudinary* sustav. Osim toga sadrži i attribute za dvije različite adrese: *url* i *thumbnailUrl*, gdje svaka adresa usmjerava na jednu od fotografija dobivenih korištenjem dva navedena skupa transformacija, gdje prva usmjerava na kopiju prilagođenu za uvećani prikaz, dok druga na kopiju vrlo male veličine.

### 5.3. Upravljanje oglasom

Upravljanje oglasom predstavlja jednu od ključnih skupina funkcionalnosti koje aplikacija mora imati. Pod ovim skupom spadaju sve funkcionalnosti vezane uz kreiranje, upravljanje i ažuriranje oglasa. Osim toga ovaj skup funkcionalnosti obuhvaća i upravljanje rezervacijama.

#### 5.3.1. Kreiranje oglasa

Postupak kreiranja oglasa zahtjeva od najmodavca unošenje veće količine podataka. Iz tog razloga razvijen je sustav za kreiranje oglasa u više koraka kako bi se olakšao proces kreiranja oglasa. Osim toga radi poboljšanja korisničkog iskustva svaki korak sprema se u bazu podataka, što omogućava da najmodavac u bilo kojem trenutku može zatvoriti aplikaciju te nastaviti proces kreiranja oglasa u neko drugo vrijeme.

U bazu podataka dodatno se spremaju podaci koji predstavljaju trenutno stanje oglasa, te naziv zadnje uspješno izvršenog koraka. Stanje oglasa sprema se unutar baze podataka u obliku *enum* tipa podatka, time se ograničavaju moguće vrijednosti stanja oglasa na vrijednosti definirane u tablici 5.5. Na sličan način sprema i trenutni korak izrade oglasa.

Tablica 5.5. Moguća stanja oglasa.

Stanje oglasa	Enum vrijednost	Opis
draft	DRAFT	Označava oglas u izradi
approved	APPROVED	Označava aktivan oglas
hidden	HIDDEN	Označava sakriven oglas.

Proces kreiranja oglasa započinje slanjem HTTP POST zahtjeva na rutu `/api/v1/create-a-listing`. Sustav izvršava prvi korak tako da kreira novi entitet oglasa, povezuje ga s najmodavcem, te ga sprema u bazu podataka, a pozivatelju vraća identifikacijsku vrijednost oglasa *id*, generiranu prilikom spremanja u bazu podataka. Osim navedene rute, u tablici 5.6. prikazane su rute za sve ostale korake kreiranja oglasa.

Svaki idući korak izvršava se slanjem HTTP PATCH zahtjev uz sve potrebne podatke za pojedine korake koji su prikazani u tablici 5.7. Oznake *:id* unutar ruta, predstavljaju mjesto gdje aplikacija automatski upisuje identifikacijski broj oglasa generiran u prvom koraku.

Tablica 5.6. Popis ruta za kreiranje oglasa u više koraka.

API ruta	HTTP metoda
/api/v1/create-a-listing	POST
/api/v1/create-a-listing/:id/property-type	PATCH
/api/v1/create-a-listing/:id/title	PATCH
/api/v1/create-a-listing/:id/description	PATCH
/api/v1/create-a-listing/:id/images	PATCH
/api/v1/create-a-listing/:id/pricing	PATCH

Tablica 5.7. Popis potrebnih parametara po koracima kreiranja oglasa.

Naziv koraka	Naziv parametra	Tip podatka
Tip nekretnine	structureType	Niz znakova
Lokacija	country	Niz znakova
	city	Niz znakova
	postCode	Cijeli broj
	streetName	Niz znakova
	streetNumber	Niz znakova
	latitude	Decimalni broj
	longitude	Decimalni broj
Naslov	title	Niz znakova
Opis	description	Niz znakova
Galerija	image	Datoteka
Cijena	price	Cijeli broj



Općenito koraci se izvršavaju na slijedeći način:

1. Sustav dohvaća oglas iz baze podataka na temelju zaprimljene identifikacijske vrijednosti.
2. Sustav vrši autorizaciju, odnosno sustav provjerava ukoliko je korisnik koji je predao zahtjev ujedno i vlasnik oglasa.
3. Sustav vrši validaciju podataka.
4. Ukoliko su podaci ispravni sustav ažurira podatke unutar baze te pozivatelj zaprima HTTP OK odgovor i može nastaviti sa idućim koracima.
5. U slučaju da se radi o završnom koraku, sustav dodatno vrši validaciju svih podataka, te ažurira i stanje oglasa.
6. Konačno oglas je aktivan te ga je moguće pronaći prilikom pretraživanja smještaja.

### **5.3.2. Upravljanje i ažuriranje oglasa**

Nakon što je oglas uspješno kreiran, najmodavcu se otvaraju mogućnosti upravljanja oglasom. Glavne mogućnosti koje se otvaraju su ažuriranje oglasa uz brisanje i dodavanje novih slika. Osim toga najmodavcu se omogućava skrivanje oglasa, ali i njegovo potpuno brisanje. Tablica 5.8. prikazuje sve rute vezane uz upravljanje i ažuriranje oglasa.

Pri korištenju ruta prikazanih u tablici 5.8. oznaka „:id“ označava mjesto unutar pojedine rute gdje aplikacija automatski unosi identifikacijski broj oglasa.

Sustav tada putem predanog identifikatora dohvaća željeni oglas iz baze podataka. Identifikacijske vrijednosti svih korisnikovih (ukoliko se radi o najmodavcu) oglasa aplikacija dohvaća slanjem HTTP GET zahtjeva na rutu `/api/v1/me/listings`.

Tablica 5.8. Popis ruta za upravljanje i ažuriranje oglasa

API ruta	HTTP metoda	Opis
/api/v1/listings/:id	PATCH	Ažuriranje podataka pojedinog oglasa.
/api/v1/listings/:id/images	GET	Dohvaćanje popisa slika pojedinog oglasa.
/api/v1/listings/:id/images	POST	Dodavanje slike u galeriju oglasa.
/api/v1/listings/:id/images/:imgId	DELETE	Brisanje slike iz galerije oglasa.
/api/v1/listings/:id/unpublish	PATCH	Privremeno sakrivanje oglasa.
/api/v1/listings/:id/publish	PATCH	Ponovno prikazivanje oglasa.
/api/v1/listings/:id	DELETE	Brisanje pojedinog oglasa.

Nakon dohvaćanja oglasa, sustav provjerava ukoliko je pozivatelj rute autoriziran za trenutnu radnju. U kontekstu operacija upravljanja oglasom, provjera autorizacije se svodi na jednostavnu provjeru jednakosti između *host* atributa unutar dohvaćenog entiteta oglasa, te entiteta pozivatelja dohvaćenog prilikom autentifikacijske provjere, odnosno provjere JWT (*engl. JSON Web Token*) tokena. Nakon toga sustav izvršava željenu radnju, te najmodavcu vraća odgovor.

Prilikom ažuriranja podataka, najmodavac unutar zahtjeva predaje jedan ili više parametara definiranih u tablici 5.1. Parametri, odnosno podaci za ažuriranje predaju se u obliku JSON kodiranog objekta [19]. Sustav dekodira zaprimljene podatke, vrši njihovu validaciju prema unaprijed definiranim pravilima, te konačno ažurira podatke unutar baze podataka. U slučaju uspješno izvršene radnje, sustav najmodavcu vraća HTTP OK odgovor.

Osim ažuriranja podataka oglasa najmodavcu se nude mogućnosti brisanja oglasa slanjem HTTP DELETE zahtjeva na rutu */api/v1/listings/:id*. Ova radnja potpuno briše sve podatke vezane uz oglas kao i sve povezane slike. Kako je ova operacija konačna, dodana je mogućnost privremenog sakrivanja oglasa slanjem zahtjeva na */api/v1/listings/:id/unpublish*. Sakriveni oglasi se neće prikazivati prilikom pretraživanja oglasa ali će biti dostupni njegovom vlasniku. Poništavanje ove radnje vrši se slanjem zahtjeva na rutu */api/v1/listings/:id/publish*.

Operacija dodavanja slika izvršava se slanjem HTTP POST zahtjeva na rutu */api/v1/listings/:id/images*, pri čemu je potrebno sliku predati u odgovarajućem obliku i formatu.

Slanjem HTTP GET zahtjeva na rutu */api/v1/listings/:id/images*, moguće je dohvatiti popis slika vezanih uz pojedini oglas. Unutar popisa slika vraćene su i identifikacijske vrijednosti pojedinih slika, te u kombinaciji s rutom */api/v1/listings/:id/images/:imgId* omogućeno je brisanje pojedinih slika.

### 5.3.3. Upravljanje rezervacijama

Mogućnost upravljanja rezervacijama ključna je iz razloga što omogućava najmoprimcima jednostavan pronalazak smještaja koji su dostupni unutar nekog željenog perioda najma. U tablici 5.9. prikazane su sve rute vezane uz koncept rezervacija. Također atributi modela rezervacije definirani su u tablici 5.10.

Tablica 5.9. Popis ruta za upravljanje rezervacijama

API ruta	HTTP metoda	Opis
<i>/api/v1/listings/:id/reservations</i>	GET	Dohvaćanje popisa rezervacija pojedinog oglasa
<i>/api/v1/listings/:id/reservations</i>	POST	Dodavanje rezervacije za pojedini oglas.
<i>/api/v1/listings/:id/reservations/:resId</i>	GET	Dohvaćanje pojedine rezervacije.
<i>/api/v1/listings/:id/reservations/:resId</i>	PUT	Ažuriranje pojedine rezervacije.
<i>/api/v1/listings/:id/reservations/:resId</i>	DELETE	Brisanje pojedine rezervacije.

Rezervacijama može isključivo upravljati vlasnik oglasa. U dogovoru s najmoprimcem, vlasnik smještaja kreira period najma slanjem HTTP POST zahtjeva na rutu */api/v1/listings/:id/reservations*, zajedno sa svim potrebnim podacima koji su definirani u tablici 5.10. Podaci o datumima prenose se u obliku niza znakova formata *yyyy/mm/dd*.

Na temelju vrijednosti oznake *:id* sustav dohvaća potreban oglas, te nakon uspješne autorizacije i validacije predanih podataka, sustav generira entitet rezervacije, povezuje ga s oglasom te ga

konačno sprema u bazu podataka. Ukoliko je rezervacija uspješno spremljena, sustav najmodavcu vraća HTTP OK odgovor.

Slanjem zahtjeva na `/api/v1/listings/:id/reservations/:resId`, gdje `:resId` predstavlja identifikacijsku vrijednost rezervacije, najmodavac može ažurirati datume početka i kraja perioda najma slanjem HTTP PUT zahtjeva na navedenu rutu s istim parametrima kao i kod kreiranja nove rezervacije. Također najmodavac može i obrisati pojedinu rezervaciju slanjem HTTP DELETE zahtjeva na istu tu adresu.

Tablica 5.10. Model entiteta rezervacije.

Naziv parametra	Tip podatka	Opis
id	Cijeli broj	Predstavlja identifikacijski broj rezervacije.
listing_id	Cijeli broj	Predstavlja identifikacijski broj oglasa na koju se odnosi rezervacija.
startDate	Niz znakova	Predstavlja datum početka perioda najma. Parametar datuma je niz znakova formata yyyy/mm/dd.
endDate	Niz znakova	Predstavlja krajnji datum perioda najma. Parametar datuma je niz znakova formata yyyy/mm/dd.

Osim toga najmodavac može dohvatiti pojedinu rezervaciju slanjem HTTP GET zahtjeva na rutu `/api/v1/listings/:id/reservations/:resId`, ili može dohvatiti sve rezervacije vezane uz neki oglas slanjem zahtjeva na rutu `/api/v1/listings/:id/reservations`

#### 5.4. Pretraživanje smještaja

Za potrebe pretraživanja smještaja koristi se ruta prikazana u tablici 5.11. Navedena ruta koristi takozvane parametre pretrage (*eng. query string*), koji predstavlja niz parova ključ-vrijednost. Standard RFC 3986 [20] definira parametre pretrage kao niz znakova koji se nadovezuju na neku rutu te imaju simbol „?“ kao početak.

Iako standard ne definira separator između parova ključ-vrijednost, simbol „&“ uzima se kao standard.

Parametri pretrage nadovezuju po shemi (5-1).

$$/search?param_1 = 1& param_2 = 2& \dots &param_i = i& \dots &param_n = n \quad (5-1)$$

gdje  $param_i$  predstavlja ključ, a  $value_i$  vrijednost na mjestu ključa.

U slučaju da neki parametar  $i$  može zaprimiti više vrijednosti koristi se shema (5-2).

$$/search?param_i[] = 1&param_i[] = 2& \dots &param_i[] = n \quad (5-2)$$

gdje je dekodirana vrijednost  $i$ -tog parametra, odnosno ključa polje s vrijednostima 1,2, ...,  $n$ .

Tablica 5.11. Ruta za pretraživanje smještaja

API ruta	HTTP metoda
/api/v1/search	GET

U tablici 5.12. opisani su mogući ključevi uz odgovarajuće tipove podataka. Prilikom pretraživanja jedini nužan parametar je *poi* (eng. *point-of-interest*), koji predstavlja koordinate neke točke u prostoru. Tim parametrom pretragu se ograničava na smještaje koji se nalaze unutar kružnice s centrom u točki definiranoj parametrom *poi* i polumjerom definiranim parametrom.

Tablica 5.12. Popis mogućih parametara prilikom pretrage smještaja.

Naziv parametra	Tip podatka	Opis	Nužno
amenities	Polje brojevnih vrijednosti	Omogućava pretragu po pogodnostima. Predstavlja niz identifikatora pogodnosti.	Ne
poi	Niz znakova	Omogućava pretragu po blizini neke točke. Format koordinata je kombinacija geografske duljine i širine: [ <i>širina, duljina</i> ]	Da
radius	Cijeli broj	Predstavlja radius pretrage u okolini točke definirane parametrom <i>poi</i> .	Ne
price_min	Cijeli broj	Ograničava pretragu na smještaje s cijenom najma većom od predane vrijednosti.	Ne
price_max	Cijeli broj	Ograničava pretragu na smještaje s cijenom najma manjom od predane vrijednosti.	Ne
from	Niz znakova	Ograničava pretragu na smještaje koji su slobodni s početkom predane vrijednosti. Format datuma je <i>mm/dd/yyyy</i>	Ne
to	Niz znakova	Koristi se u kombinaciji s parametrom <i>from</i> . Ograničava pretragu na smještaje koji su slobodni unutar perioda definiranog predanim vrijednostima. Format datuma je <i>mm/dd/yyyy</i>	Ne

### 5.4.1. Generiranje SQL upita

Za potrebe pretraživanja smještaja razvijena je klasa *SearchQueryBuilder*, koja na temelju parametara pretrage generira odgovarajući SQL upit. Prilikom zaprimanja zahtjeva, Symfony sustav generira objekt klase *Request* koji predstavlja objekt trenutnog zahtjeva. Taj objekt sadrži brojne informacije o trenutnom zahtjevu, koji između ostalog sadrži i parametre pretrage u obliku polja.

Parametri pretrage predaju se objektu klase *SearchQueryBuilder*, a pozivanjem funkcije *apply*, čiji je kod dan na slici 5.4., generira se SQL upit. Navedena funkcija gradi SQL upit na način da iterira kroz sve parametre pretrage koji su prisutni u predanom zahtjevu, te za svaki parametar poziva odgovarajuću funkciju koja realizira jedan dio filtra.

```
public function apply(): void
{
    foreach ($this->all() as $name => $value) {
        $method = str_replace('_', '', lcfirst(ucwords($name, '_')));

        if (method_exists($this, $method) && is_callable([$this, $method])) {
            $this->$method($value);
        }
    }
}
```

Slika 5.4. *apply* funkcija za realizaciju filtra

Na slici 5.5. dan je primjer funkcije koja postavlja filter maksimalne cijene najma. Zbog načina na koji *apply* funkcija radi, nužno je da naziv funkcije bude u skladu s nazivom parametra.

U ovom kontekstu, naziv parametra je *price\_max*, stoga je potrebno da funkcija ima naziv *priceMax*, odnosno potrebno je naziv parametra pretvoriti u tzv. *Camel Case* format gdje se svaka riječ u nazivu osim prve piše velikim početnim slovom dok se brišu razmaci.

```

private function priceMax($value): void
{
    $this->qb
        ->andWhere($this->qb->expr()->lt('l.price', ':priceMax'))
        ->setParameter('priceMax', $value);
}

```

Slika 5.5. *priceMax* funkcija za ograničavanje cijene najma prilikom pretrage

Prikazana funkcija za postavljanje maksimalne cijene koristi standardnu WHERE klauzulu kako bi omogućila dohvaćanje samo onih oglasa čija je cijena najma manja od vrijednosti predane putem parametra pretrage. Većina ostalih funkcija, s iznimkom funkcije za pretraživanje po lokaciji, realizira dijelove filtra koristeći slične standardne SQL klauzule.

Funkcija *poi* koja realizira dio filtra koji omogućava pretraživanje po lokaciji nešto je kompleksnija. Kod za navedenu funkciju dan je na slici 5.6.

```

private function poi($point): void
{
    list($lat, $long) = sscanf($point, '%f,%f');

    $this->qb
        ->addSelect("ST_Distance(l.location.coordinates, ST_GeomFromText(:point, 4326), 'kilometre') AS distance")
        ->andHaving('distance <= :radius')
        ->setParameter('point', sprintf('Point(%f %f)', $lat, $long))
        ->addOrderBy('distance');

    if (!array_key_exists('radius', $this->all())) $this->qb->setParameter('radius', 1);
}

```

Slika 5.6. *poi* funkcija za pretragu po lokaciji

Kako se koordinate predaju funkciji u obliku niza slova, funkcija najprije vrši konverziju u decimalni zapis. Nakon toga se koriste dvije SQL naredbe: SELECT i HAVING, kako bi se realizirala pretraga po udaljenost. Konačno se postavlja vrijednost parametra *radius* čime se pretraga ograničava na neko područje.

Korištenjem SELECT naredbe, formira se novi stupac koji sadržava izračunate udaljenosti između predane koordinate i koordinate lokacije smještaja, a primjenom HAVING naredbe ograničava se rezultat pretrage na one smještaje čija je udaljenost manja od vrijednosti parametra *radius*.



Za izračun udaljenosti ključne su dvije ugrađene SQL funkcije:

- *ST\_GeomFromText*: vrši konverziju između tekstualnog WKT (*Well-Known-Text*) zapisa koordinate u njezin binarni zapis.
- *ST\_Distance*: uzima dvije točke u binarnom obliku te kao rezultat daje njihovu međusobnu udaljenost u odabranoj mjernoj jedinici.

#### 5.4.2. Straničenje rezultata pretrage

Prilikom pretraživanja smještaja postoji mogućnost potencijalno velikog broja rezultata. Kako bi se spriječilo preopterećenje sustava, te poboljšalo korisničko iskustvo nužno je ograničiti broj rezultata koji se prikazuju.

U te svrhe koristi se metoda straničenja koja funkcionira na način da rezultate pretrage podijeli u manje skupine koje se nazivaju *stranice*. Stranice ograničavaju broj rezultata prikazanih korisniku u jednom trenutku, a parametrom *n*, koji se naziva *veličina stranice* ograničava se broj rezultata stranice.

Rute koje implementiraju navedenu metodu koriste dodatni parametar pretrage koji se naziva *page* i predstavlja redni broj stranice putem kojeg se omogućava pomicanje kroz rezultate pretrage.

Najjednostavnija metoda straničenja je metoda *offset*, koja se temelji na upotrebi LIMIT i OFFSET naredbi. Primjenom LIMIT naredbe, ograničava se broj rezultata, odnosno postavlja je veličina stranice, dok naredba OFFSET omogućava preskakanje rezultata kako bi se došlo do željene stranice.

Glavna prednost, ali ujedno i nedostatak ove metode je korištenje OFFSET naredbe. Primjenom ove naredbe moguće je dohvatiti stranice proizvoljnog rednog broja, međutim izvršavanje ove naredbe je neefikasno iz razloga što baza podataka mora iterirati kroz sve prethodne stranice.

Drugo popularno rješenje je korištenje *keyset* metode, pri čemu se rješava problem efikasnosti prijašnje metode međutim onemogućuje se preskakanje stranice pa je potrebno sve redom učitati dok se ne dođe do željene stranice.

U kontekstu ove aplikacije, preskakanje stranica predstavlja korisnu funkcionalnost. Zbog toga odabrana je metoda straničenja rezultata pretrage koja se temelji na LIMIT/OFFSET.

## 6. RAZVOJ KORISNIČKOG SUČELJA

SPA (*engl.* Single Page Application) aplikacije pružaju alternativu nad tradicionalnim aplikacijama koje prikazuju sadržaj kroz više stranica. Glavna karakteristika SPA pristupa je da navigacija između različitih dijelova aplikacije ne zahtjeva potpuno ponovno učitavanje traženog sadržaja. Umjesto toga, korisničko sučelje dinamički se generira i ažurira novim sadržajem.

Velika prednost SPA pristupa je da omogućava razvoj klijentske aplikacije koja je potpuno neovisna o detaljima implementacije i o tehnologijama koje su korištene prilikom razvoja poslužiteljskog dijela. Na taj način poslužitelj se može posvetiti isključivo obradi i prijenosu podataka. Osim toga ovaj pristup omogućava paralelno razvijanje klijentske i poslužiteljske aplikacije što ubrzava proces razvoja.

Postoje brojne JavaScript biblioteke poput *React*, *Angular* i *Vue*, koje olakšavaju razvoj takvih oblika web aplikacija. Osim navedenih biblioteka, vrlo važna tehnologija koja se primjenjuje prilikom razvoja je AJAX (*engl.* Asynchronous JavaScript and XML). AJAX tehnologija omogućava asinkronu komunikaciju između klijentskog i poslužiteljskog dijela aplikacije.

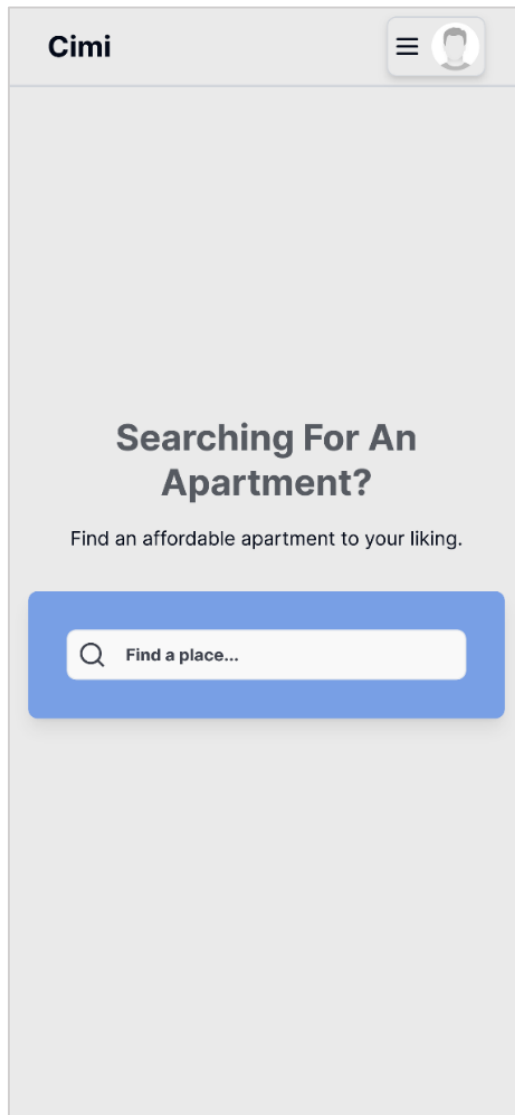
Prilikom korištenja aplikacije, korisnik nije svjestan komunikacije jer se ona odvija u pozadini bez njegovog znanja. Na taj način se ostvaruje dojam responzivnosti i kratkog odziva.

Primjena ovog pristupa značajno poboljšava korisničko iskustvo, te omogućava razvoj responzivnog i dinamičnog korisničkog sučelja koje ima kratak odziv.

### 6.1. Početna stranica

Na slici 6.1. prikazana je početna stranica web aplikacije. Početna stranica sastoji se od alatne trake koja sadrži glavni izbornik a na čiji pritisak se otvara modalni prozor koji prikazuje korisničke mogućnosti. U slučaju da korisnik nije prijavljen, glavni meni korisniku nudi mogućnost prijave ili registracije novog računa.

Početna stranica ujedno služi kao i početna točka pri pretraživanju smještaja pa sadrži i tipku „Find a place“ putem koje se započinje postupak pretrage smještaja.

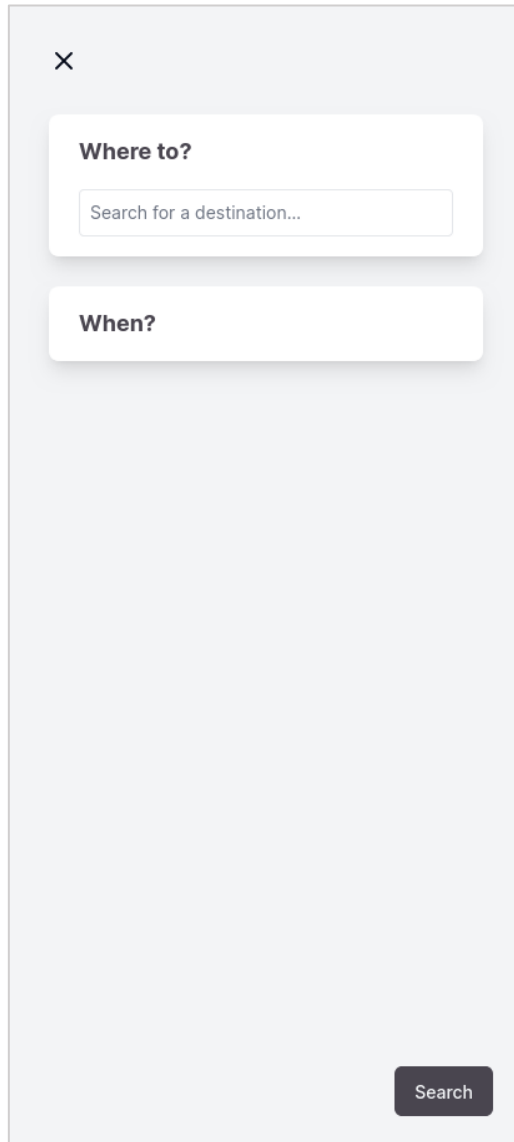


Slika 6.1. Početna stranica web aplikacije.

## 6.2. Pretraživanje smještaja

Najmoprimac započinje postupak pretraživanja smještaja pritiskom na tipku „Find a place“ koja se nalazi na početnoj stranici. Pritiskom na navedenu tipku otvara se modalni prozor za pretraživanje oglasa koji je prikazan na slici 6.2.

Modalni prozor prikazuje dva elementa za unos podataka, prvi od kojih predstavlja komponentu za pretraživanje lokacija, dok drugi predstavlja komponentu kalendara za izbor perioda najma. Otvaranjem prozora pretrage najmoprimcu se automatski otvara prozor za pretragu lokacije, a nakon potvrde izbora lokacije automatski se otvara prozor za odabir perioda najma. Konačno na dnu stranice nalazi se tipka „Search“ kojom se izvršava pretraga smještaja te se najmoprimca usmjerava na stranicu sa rezultatima.



Slika 6.2. Komponenta za pretragu smještaja.

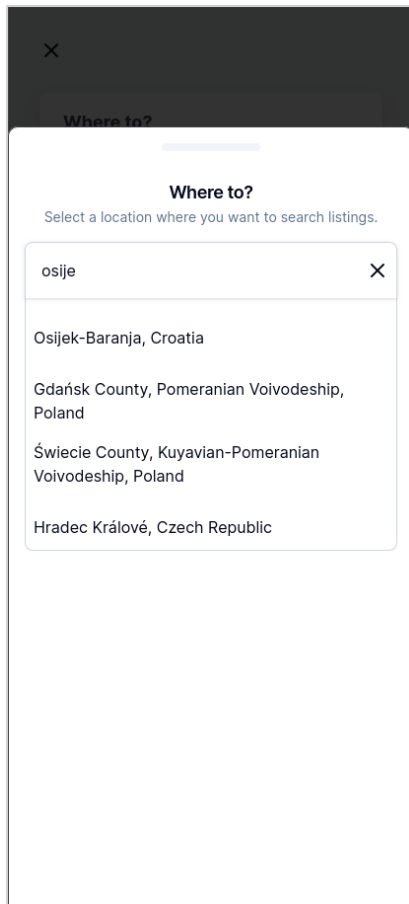
Komponenta izbora lokacije sadrži element za unos teksta, a pritiskom na taj element otvara se modalni prozor koji prikazan na slici 6.3. a), unutar kojeg najmpriamac pretražuje lokacije putem njihovog naziva.

Za potrebe pretrage koristi se *Mapbox* [24] usluga koja omogućava automatsko predlaganje lokacije na temelju unesenog teksta, te omogućava pristup koordinatama lokacije.

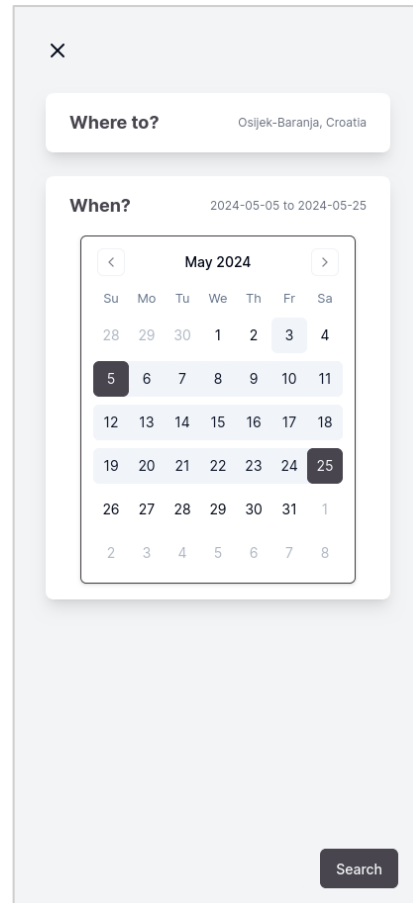
Odabirom jedne od ponuđenih lokacija, modalni prozor se zatvara a zatim se otvara idući element u prozoru koji je prikazan na slici 6.3. b). Unutar tog elementa najmpriamac odabire željeni period

najma. Pretraga dohvaća samo oglase koji su slobodni unutar tog odabranog perioda u blizini željene lokacije.

Nakon odabira lokacije i perioda najma, najmodavac pritišće na tipku „Search“ putem koje aplikacija izvršava pretragu koristeći odabrane parametre.



Slika 6.3. a) Modalni prozor za odabir lokacije.

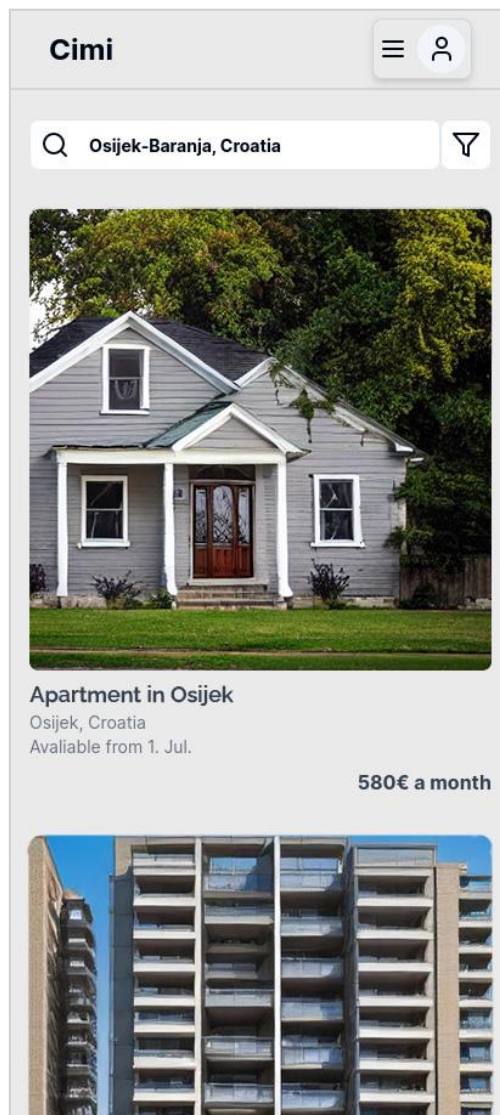


Slika 6.3. b) Komponenta za odabir datuma.

Filter komponenta također sadrži takozvani kontekst. Kontekst je posebna komponenta unutar *React* razvojnog okvira koja pruža mogućnosti prosljeđivanja podataka između komponenata unutar tog konteksta. Tipično se u takvim scenarijima definira i funkcija koja dohvaća određeni kontekst.

Kako bi se očuvalo stanje filtera između različitih stranica web aplikacije razvijena je komponenta *FilterContext* za uspostavljanje konteksta, te funkcija *useFilter* koja dohvaća navedeni kontekst. Konkretno funkcija koja dohvaća kontekst vraća strukturu sa svim parametrima filtra, te dvije funkcije: *updateParams* koja ažurira parametre i *handleSearch* koja izvršava pretragu i navodi najmoprimca na stranicu s rezultatima pretrage.

### 6.2.1. Stranica sa rezultatima pretrage



Slika 6.4. Stranica rezultata pretrage.

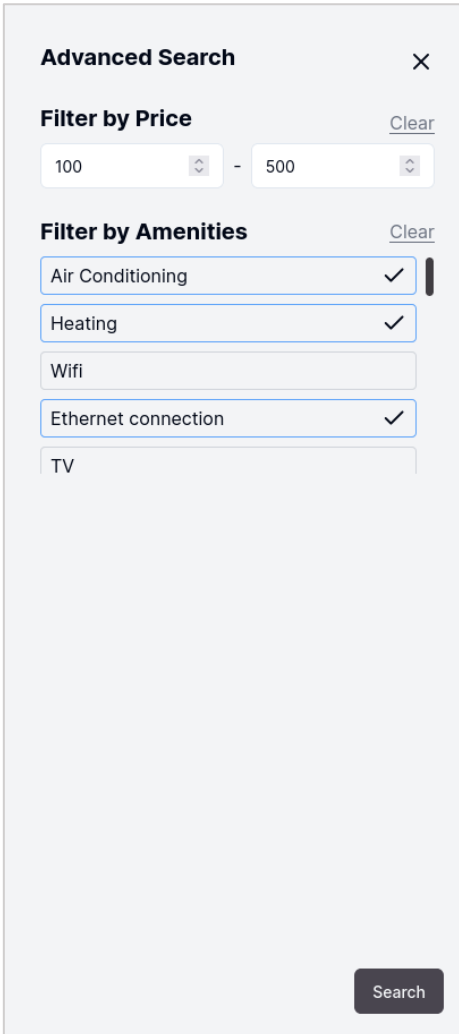
Na slici 6.4. prikazana je stranica sa rezultatima pretrage. Rezultati pretrage oblikovani su u obliku popisa. Elementi tog popisa prikazuju osnovne informacije o smještaju poput naziva, lokacije, najranijeg datuma i cijena najma. Ujedno svaki od tih elemenata ponaša se kao i poveznica koja usmjerava na odgovarajuću stranicu oglasa.

Rezultati pretrage prikazuju se u obliku stranica, odnosno prilikom pretrage ne vraćaju se svi rezultati. Kontrole za upravljanje trenutnom stranicom nalaze se na dnu stranice te se sastoje od dvije tipke, gdje jedna tipka učitava prethodnu stranicu, dok druga učitava iduću.

Osim toga pri početku stranice ponovno je prikazana filter komponenta, te dodatno i tipka za detaljnije filtriranje.

### 6.2.2. Napredno filtriranje rezultata

Pritiskom na tipku za detaljno filtriranje otvara se novi modalni prozor komponente za detaljno filtriranje. Modalni prozor prikazan je na slici 6.5. Navedeni prozor sadrži dodatne postavke za filtriranje poput cijene najma te nudi mogućnost filtriranja po pogodnostima. Nakon odabira dodatnih filtra, pritiskom na tipku „Search“ ponovno se izvršava pretragu a popis se ažurira novim rezultatima pretrage.



The image shows a modal window titled "Advanced Search" with a close button (X) in the top right corner. It is divided into two main sections: "Filter by Price" and "Filter by Amenities".

- Filter by Price:** This section includes a "Clear" link on the right. Below it are two input fields with dropdown arrows, containing the values "100" and "500", separated by a minus sign.
- Filter by Amenities:** This section includes a "Clear" link on the right. It contains a list of amenities, each with a checkbox and a checkmark:
  - Air Conditioning ✓
  - Heating ✓
  - Wifi
  - Ethernet connection ✓
  - TV

A "Search" button is located at the bottom right of the modal.

Slika 6.5. Komponenta za napredno filtriranje rezultata pretrage.

### 6.3. Stranica oglasa

Stranica oglasa sadrži veći broj komponenti koje najmoprimcima prikazuju detaljne informacije o smještaju. Na slici 6.6. možemo vidjeti primjer stranice jednog oglasa.

Pri početku stranice oglasa nalazi se naslov oglasa i komponenta galerije koja sadrži fotografije smještaja. Nakon toga stranice sadrži komponentu koja prikazuje informacije o vrsti i lokaciji smještaja, te daljnji opis smještaja koji može sadržavati dodatne informacije poput kvadrature, broja prostorija i ostale informacije koje najmodavac smatra korisnim. U slučaju da opis smještaja ne stane u za to predviđenu lokaciju, tekst se skraćuje te se prikazuje tipka na čiji se pritisak otvara modalni prozor koji sadrži cijeli tekst opisa smještaja.

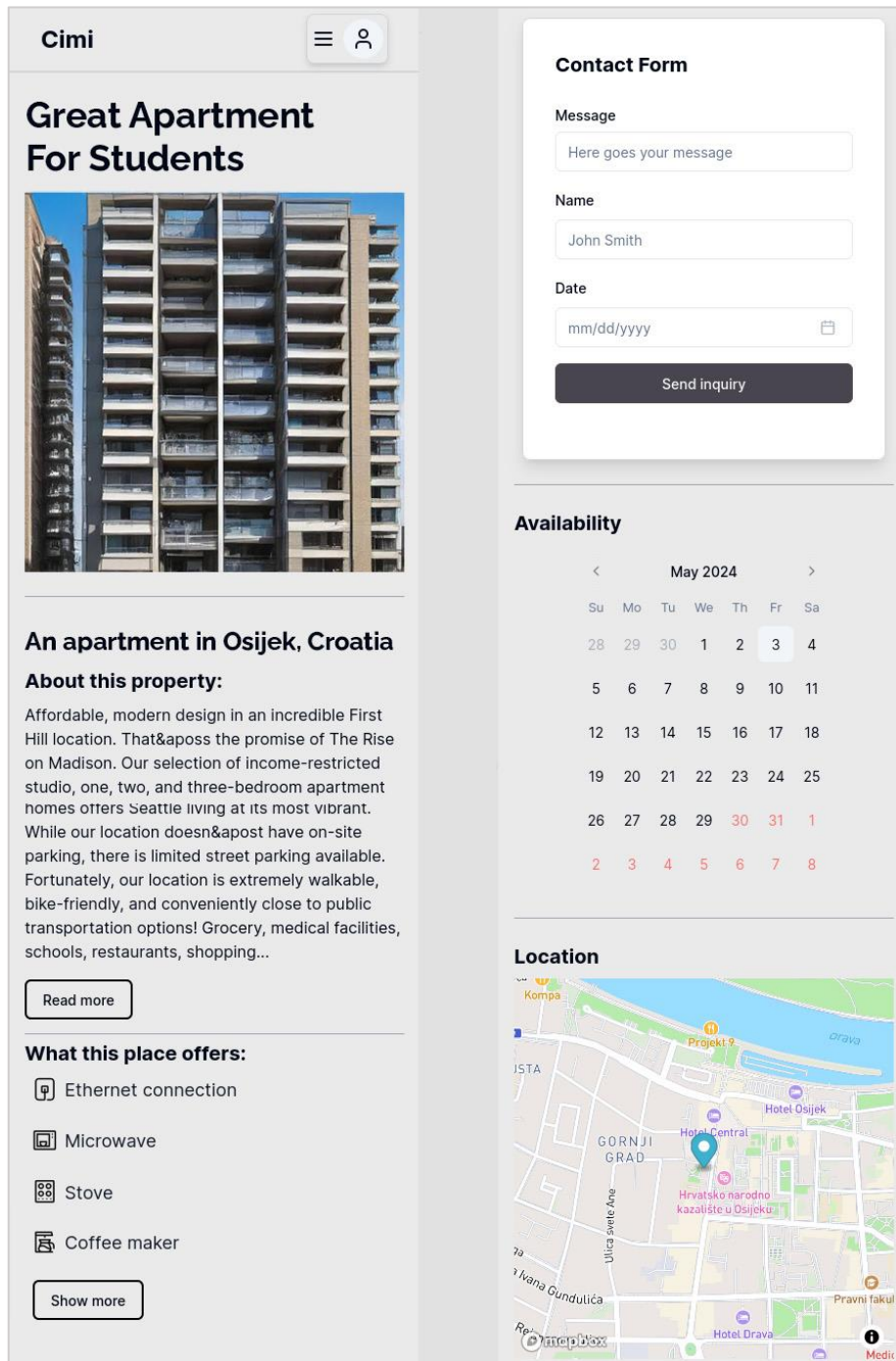
Nadalje stranica sadrži komponentu koja prikazuje popis svih pogodnosti koje nudi smještaj, gdje se također u slučaju velikog broja pogodnosti prikazuje tipka na čiji se pritisak otvara modalni prozor koji sadrži popis svih pogodnosti koje nudi taj smještaj.

U nastavku te komponente prikazuje se obrazac putem kojeg najmoprimac može započeti razgovor s najmodavcem, odnosno ta forma je predviđeni prvi korak u procesu iznajmljivanja smještaja. Putem obrasca najmoprimac unosi informacije o vlastitom imenu, željenom datumu useljenja, te početnu poruku. Pritiskom na tipku „*Send Inquiry*“, otvara se komunikacijski kanal između najmoprimca i najmodavca.

Nakon obrasca, stranica prikazuje komponentu s kalendarom, gdje najmoprimac može pregledati periode unutar kojih je smještaj slobodan. Unutar kalendara, dani označeni crvenom bojom predstavljaju dane unutar kojih je nekretnina zauzeta. Osim toga na vrhu komponente nalaze sve dvije tipke putem kojih se najmoprimac može kretati između različitih mjeseca.

Konačno stranica prikazuje kartu na kojoj je simbolom označena točna lokacija smještaja. Uz to, omogućeno upravljanje kartom, a budući da karta prikazuje i ostale objekte u blizini smještaja, najmoprimac može istražiti okolinu i proučiti objekte od interesa poput različitih trgovina i slično.





Slika 6.6. Razlomljeni prikaz stranice oglasa u prilagođen za prikaz na mobilnom uređaju.

## 6.4. Obrazac za izradu oglasa u više koraka

Obrazac za izradu oglasa sastoji se od većeg broja koraka gdje najmodavac u svakom koraku unosi samo jedan dio informacija o smještaju. Svaki korak se sprema u bazu podataka pozivanjem definiranih ruta. Na taj način najmodavcu se omogućava nastavak izrade oglasa u neko drugo vrijeme ne gubeći pri tome postignuti napredak.

Postoji nekoliko biblioteka koje implementiraju sustav za obrasce u više koraka, međutim testirane biblioteke ne omogućavaju ili ne pružaju jednostavan način za spremanje pojedinačnih koraka, stoga je bilo potrebno razviti vlastiti sustav.

Zbog nešto veće kompleksnosti ovog načina implementacije obrasca na klijentskoj strani, izabrana je tehnika konačnih automata kako bi se pojednostavilo upravljanje stanjem sustava za izradu oglasa. Osim toga unutar *React* okruženja postoji nekoliko biblioteka koje olakšavaju rad s konačnim automatima, a odabrana je biblioteka pod nazivom *XState* [21].

Navedena biblioteka omogućava jednostavno definiranje skupa stanja te događaja koji uzrokuju prijelaz iz jednog stanja u drugo.

Najmodavac može pokrenuti proces kreiranja oglasa pritiskom na tipku za kreiranje oglasa ili može nastaviti postupak izrade odabirom oglasa koji se prikazuje unutar kućice s nezavršenim oglasima. Nastavno na to, aplikacija navodi najmodavca na stranicu za izradu oglasa, a sustav ulazi u početno stanje *loadingDraft* gdje se učitavaju do sada uneseni podaci.

Slijedeće stanje je *redirecting* gdje se vrši provjera parametra *lastUpdatedStep* učitano u prethodnom stanju. Navedeni parametar predstavlja zadnje ažurirano stanje te ovisno o vrijednosti tog parametra, unutar automata, odašilje se događaj koji uzrokuje prijelaz u odgovarajuće stanje.

Svako iduće stanje koje odgovara jednom od koraka funkcionira na isti princip. Svako stanje definira dva događaja:

- Događaj *back* uzrokuje prijelaz u prethodno stanje, odnosno prethodni korak u izradi oglasa. Izuzetak je stanje prvog koraka koje ne definira ovaj događaj.
- Događaj *next* uzrokuje prijelaz u iduće stanje, odnosno idući korak u izradi oglasa.

Generiranje ovih događaja provodi se izvan konačnog automata, odnosno unutar obrazaca svakog koraka nalaze se tipke kojima se najmodavac može pomicati između koraka i koje odašilju odgovarajuće događaje.

Svaki obrazac prikuplja određeni jedan dio podataka te ih prosljeđuje na odgovarajuću rutu, a u slučaju pogreške obrazac provodi oporavak od pogreški. Ukoliko je korak uspješno proveden obrazac generira događaj *next* koji uzrokuje prijelaz u iduće stanje. Osim toga, u slučaju pritiska na tipku koja navodi na prethodni korak, sustav jednostavno generira događaj *back* koji uzrokuje prijelaz u stanje koje odgovara prethodnom koraku. Ovim je opisan način djelovanja konačnog automata.

Kako bi sustav ispravno funkcionirao potrebno je prikazati odgovarajući obrazac. To je postignuto kodom prikazanim na slici 6.7. koji definira stranicu koraka, te je smješten unutar dinamičke rute `/wizard/create-a-listing/[id]/[step]`.

```
function Page({ params }: { params: { id: string; step: string } }) {
  const state = WizardMachineContext.useSelector((s) => s);
  const router = useRouter();

  useEffect(() => {
    if (!state?.matches('loadingDraft') && !state.matches('loaded')) {
      router.push(`/create-a-listing/${params.id}/${state.value}`);
    }
  }, [state, params.id, router]);

  return <div>{params.step === state.value && Forms[state.value]}</div>;
}
```

Slika 6.7. Kod stranice koraka pri izradi oglasa.

Korištenjem `useEffect` funkcije definiraju funkcionalnosti koje se izvršavaju svaki put kada dođe do promjene u nekoj od promatranih vrijednosti. U ovom slučaju procedura se izvršava ukoliko dođe do promjene stanja konačnog automata.

Pri promjeni stanja procedura poziva `router.push` metodu koja izvršava navigaciju na rutu predanu toj funkciji. Kako se radi o dinamičkoj ruti i navigacija navodi na istu stranicu dolazi do promjene URL adrese. Konkretno unutar rute mijenja se isključivo vrijednost `step` parametra koji definira trenutni korak u izradi oglasa. Konačno prikazana funkcija koja definira stranicu, na temelju te vrijednosti, vraća komponentu obrasca koja odgovara trenutnom koraku.

Za izradu obrasca korištena je *React Hook Forms* [22] biblioteka. Navedena biblioteka korisna je iz razloga što omogućava vrlo jednostavno upravljanje stanjem obrasca, uključujući praćenje unosa, provjeru valjanosti podataka, upravljanje greškama i rukovanje podnošenja obrasca. Osim toga ima vrlo dobru integraciju sa *Zod* [23] bibliotekom koja omogućava definiranje sheme za validaciju podataka, odnosno omogućava da se za svaki atribut definira tip podataka, raspon vrijednosti, minimalan i maksimalan broj znakova itd.

**Title**

Chose a title for your property...

This is a public title for your property listing

Previous Next Save

Slika 6.8. a) Obrazac za odabir vrste smještaja.

**Description**

Chose a description for your property...

This is a public description for your property listing

Previous Next Save

Slika 6.8. b) Obrazac za odabir pogodnosti.

**Which of these best describes the type of your property?**

House Apartment

Previous Next Save

Slika 6.8. c) Obrazac za postavljanje fotografija.

**Select multiple amenities that your property provides.**

You need to select at least 3 amenities.

Air Conditioning Heating

Wifi Ethernet connection

TV Refrigerator

Microwave Dishes and silverware

Dishwasher Stove

Oven Coffee maker

Cooking basics Bed linens

Hairdryer Washing Machine

Previous Next Save

Slika 6.8. d) Obrazac za upis naslova.

**Where is your property located?**  
Use search to find the location of your property.

Search

Previous Next Save

Slika 6.8. e) Obrazac za odabir lokacije smještaja.

**Where is your property located?**  
Use search to find the location of your property.

Street Name  
Vukovarska Ulica

Street Number  
8

Post Code  
44000

City  
Sisak

Country  
Croatia

Search Vukovarska Ulic ...

Previous Next Save

Slika 6.8. f) Obrazac za unos adrese smještaja.

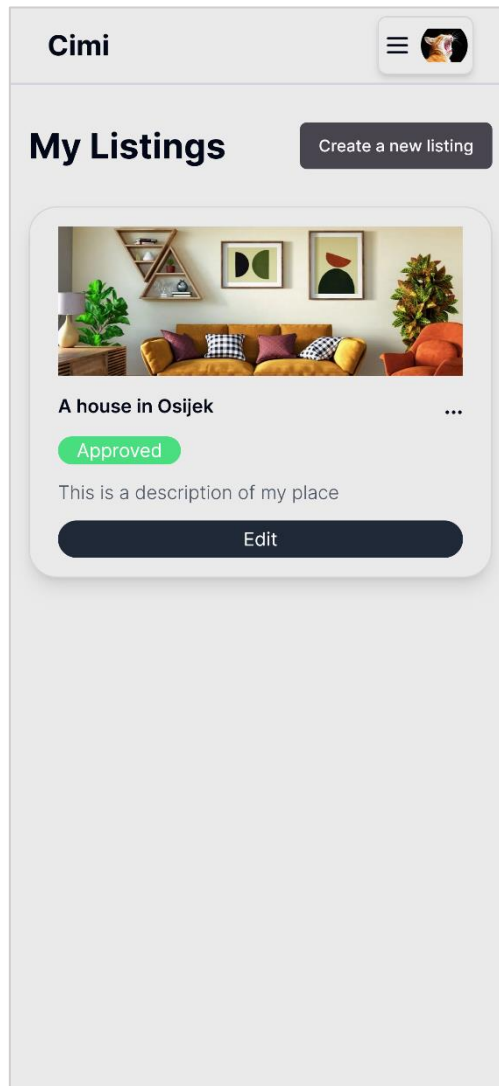
Na slikama od 6.8. a) do 6.8. e) prikazani su obrasci za pojedine korake pri izradi oglasa. Na slici 6.8. e) prikazan je korak za unos lokacije smještaja, gdje najmodavac najprije koristi funkcionalnost pretrage adresa na interaktivnoj karti, a odabirom na jedan od rezultat pretrage adresa otvara se dodatni obrazac prikazan na slici 6.8. f) gdje najmodavac može urediti detalje adrese.

## 6.5. Upravljanje oglasima

Na slici 6.9. prikazana je stranica na kojoj najmodavac ima mogućnosti upravljanja vlastitim oglasima. Prikazana stranica nudi nekoliko mogućnosti, najprije najmodavac ima mogućnost kreiranja novog oglasa pritiskom na tipku „Create a listing“.

Iduće na stranici je prikazan popis svih oglasa koje je najmodavac prethodno kreirao. Svaki pojedini prikaz oglasa omogućuje uređivanje oglasa pritiskom na tipku „Edit“ koja najmodavca navodi na stranicu za uređivanje koja funkcionira na isti princip kao i postupak kreiranja oglasa.

Pritiskom na tipku čija se ikona sastoji od tri točke najmodavac može pristupiti dodatnim mogućnostima za odabrani oglas poput pregleda stranice oglasa, dodavanje perioda najma, te brisanje oglasa.




Slika 6.9. Korisničko sučelje za upravljanje oglasima.

Pri odabiru funkcionalnosti dodavanja perioda najma najmodavcu se otvara prozor prikazan na slici 6.10. Prozor prikazuje obrazac za dodavanje perioda najma koji se sastoji od nekog početnog i nekog drugog krajnjeg datuma. Nakon unosa datum, pritiskom na tipku „Submit“ sustav kreira novi period najma te obavještava najmodavca.


**Manage Availability**

Create a new rent period for this property. Your property will be marked as occupied for the selected period.

Start Date

mm/dd/yyyy 

End Date

mm/dd/yyyy 

Submit

Cancel

Slika 6.10. Obrazac za kreiranje novog perioda najma.

## 6.1. Komunikacijski sustav i stranica s razgovorima

Za implementaciju komunikacijskog sustava korištena je usluga *GetStream* [25]. *GetStream* je platforma koja omogućava razvoj aplikacija s funkcionalnostima poput praćenja aktivnosti korisnika, komentiranja, označavanja sadržaja, korištenje kompleksnih algoritama namijenjenih za generiranje preporuka te razvoja stvarnovremenskih komunikacijskih aplikacija. *GetStream* također nudi i brojne biblioteke namijenjene za jednostavno korištenje njihovog API sustava.

U prethodnom poglavlju opisan je obrazac putem kojeg najmodavac može započeti komunikaciju s najmodavcem. Nakon što najmodavac podnese obrazac, sustav u pozadini koristi usluge *GetStream* sustava kako bi kreirao komunikacijski kanal između najmodavca i najmodavca. U

slučaju da je kanal uspješno kreiran, web aplikacija preusmjerava najmoprimca na korisničko sučelje komunikacijskog sustava koje prikazano na slici 6.11.

Svi korisnici u bilo kojem trenutku mogu pristupiti svojim komunikacijama putem glavnog izbornika sustava.

Važno je napomenuti da je komunikacijski sustav stvarno vremenski, što znači da se poruke prikazuju i dostavljaju u realnom vremenu, time osiguravajući brzu i učinkovitu interakciju među korisnicima.

Glavni prozor komunikacijskog sučelja prikazuje popis nedavno primljenih poruka i pruža korisnicima brz pristup njihovim razgovorima. Svaki razgovor je označen imenom sudionika i vremenom zaprimanja.



Slika 6.11. Korisničko sučelje komunikacijskog sustava.



## 7. ZAKLJUČAK

Cilj ovog rada je bio razviti web aplikaciju koja će olakšati studentima pronalazak smještaja, ali istovremeno razviti i okruženje unutar kojeg će najmodavci moći brzo i jednostavno ponuditi svoj smještaj u najam.

Najprije, u radu je dana analiza tržišta koja uključuje pregled preferencija studenta prilikom traženja smještaja, čime su dobiveni vrijedni uvidi u faktore koji utječu na odluke korisnika prilikom traženja smještaja. U sklopu toga, dana je i analiza već postojećih rješenja na tržištu čime su dobivene važne ideje i smjernice za razvoj ove aplikacije.

U nastavku rada analizirani su zahtjevi aplikacije, odnosno dan je pregled glavnih funkcionalnosti, te scenarija upotrebe. Ove analize omogućile su bolje razumijevanje potreba korisnika i definiranje ključnih funkcionalnosti i elemenata aplikacije.

U ostatku rada opisan je razvoj web aplikacije. Najprije, poslužiteljski dio web aplikacije razvijen je u skladu sa REST API principima, a za razvoj je korišten *Symfony* razvojni okvir. Navedeni poslužiteljski dio aplikacije implementira kompletnu poslovnu logiku, tj. upravlja korisničkim računima i oglasima, omogućava pretragu smještaja, te izvršava autentifikaciju i autorizaciju korisničkih zahtjeva. Kako bi se pohranili korisnički podaci korištena je relacijska baza podataka MySQL.

Za razvoj klijentske strane korišten je *NextJS* razvojni okvir, koji omogućava izradu dinamičkih stranica. Dok je komunikacijski sustav ostvaren putem web usluge *GetStream*, čime je ostvarena funkcionalnost koja omogućava da studenti i najmodavci stupe u kontakt. Osim toga za pohranu slika korištena je usluga *Cloudinary* koja omogućava jednostavnu pohranu i optimizaciju slika.

Prilikom razvoja korisničkog sučelja primijenjen je princip „mobile first“ čime je web aplikacije najprije prilagođena mobilnim uređajima, a zatim proširena za veće zaslone. Tim načinom osigurano je optimalno korisničko iskustvo na svim uređajima. Kako je postupak kreiranja oglasa jedna od najvažnijih funkcionalnosti i zahtjeva unos veće količine informacija izabran je princip obrasca u više koraka, gdje se svaki korak sprema u bazu podataka. Na taj način omogućeno je da najmodavac može u bilo kojem trenutku zatvoriti aplikaciju te nastaviti proces kreiranja oglasa u neko drugo vrijeme.

U trenutnom obliku, web aplikacija implementira sve osnovne funkcionalnosti kako bi osigurala ispravan rad i pružila korisnicima osnovne mogućnosti koje se očekuju od ovakve platforme.

U kontekstu poboljšanja ove aplikacije, jedno od područja vrijedno pažnje je korisničko iskustvo, posebice na strani najmodavca. Neke od ideja uključuju poboljšanje sustava za upravljanje rezervacijama, poput omogućavanja jednostavnijeg prikaza, pretraživanja i uređivanja svih rezervacija kroz integraciju s kalendarom.

Osim toga, aplikaciji nedostaje sustav automatskih notifikacija koji bi obavještavao najmodavce ali i studente o zaprimljenim porukama ali i funkcionalnost ocjenjivanja smještaja od strane najmoprimca.

Također vrlo zanimljivo poboljšanje je u pogledu omogućavanja interakcije između najmoprimaca kod istog najmodavca. Na taj način omogućio bi je jednostavniji pronalazak sustanara.

## LITERATURA

- [1] UNESCO, Paris : UNESCO, 2024, 2024. UNESCO, “Internationalizing higher education: findings from the fifth consultation the implementation of the Recommendation on the Recognition of Studies and Qualifications in Higher Education,” [online], Paris : UNESCO, 2024. Dostupno na: <https://unesdoc.unesco.org/ark:/48223/pf0000103586>. [26.7.2024.].
- [2] Eurostat, “Eurostat Education and Training Database” [online]. Dostupno na: <https://ec.europa.eu/eurostat/web/education-and-training/database>. [13.5.2024.].
- [3] Alina Żróbek-Róžańska, Lucyna Szulc, “Over a Million Student Tenants in Poland. Analysis of Preferences,” *Real Estate Management and Valuation*, no. 2, vol. 26, Jun. 2018.
- [4] Ann Verhetsel, Roselinde Kessels, Toon Zijlstra, Marjolein Van Bavel, “Housing preferences among students: collective housing versus individual accommodations? A stated preference study in Antwerp (Belgium),” *Springer Science+Business*, 2016.
- [5] Booking [online]. Dostupno na: <https://www.booking.com/>. [12.4.2024.].
- [6] Hostelworld [online]. Dostupno na: <https://www.hostelworld.com/>. [12.4.2024.].
- [7] Airbnb [online]. Dostupno na: <https://www.airbnb.com/>. [12.4.2024.].
- [8] Njuškalo [online]. Dostupno na: <https://www.njuskalo.hr/iznajmljivanje-stanova/osijek>. [29.8.2024.].
- [9] Docker [online]. Dostupno na: <https://www.docker.com/>. [12.4.2024.].
- [10] Symfony Framework, Symfony SAS [online]. Dostupno na: <https://symfony.com/>. [12.4.2024.].
- [11] React Library [online]. Dostupno na: <https://react.dev/>. [12.4.2024.].
- [12] Doctrine ORM Library [online]. Dostupno na: <https://www.doctrine-project.org/>. [12.4.2024.].
- [13] Postman [online]. Dostupno na: <https://www.postman.com/>. [12.4.2024.].
- [14] E. F., Codd, “A relational model of data for large shared data banks,” *Commun. ACM*, no. 6, vol. 13, pp. 377–387, Jun. 1970.

- [15] T., Connolly, C., Begg, *Database systems: A practical approach to design, implementation and management*. Addison-Wesley Educational: Boston, MA, 2004.
- [16] Oracle, “MySQL 8.1 Reference Manual,” 2023.
- [17] National Imagery, M., Agency, “Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems,” National Imagery and Mapping Agency, St. Louis, MO, USA, Jan. 2000.
- [18] R. Fielding., “REST: Architectural Styles and the Design of Network-based Software Architectures,” University of California, Irvine, 2000.
- [19] D., Crockford, “JSON - JavaScript Object Notation.” 2006.
- [20] T., Berners-Lee, R. T., Fielding, L. M., Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” no. 3986, Jan. 2005.
- [21] XState [online]. Dostupno na: <https://xstate.js.org/>. [6.5.2024.].
- [22] React Hook Form [online]. Dostupno na: <https://react-hook-form.com/>. [6.5.2024.].
- [23] Zod [online]. Dostupno na: <https://zod.dev/>. [6.5.2024.].
- [24] Mapbox [online]. Dostupno na: <https://www.mapbox.com/>. [6.5.2024.].
- [25] GetStream [online]. Dostupno na: <https://getstream.io/chat/>. [6.5.2024.].

## SAŽETAK

U ovom radu razvijena je web aplikacija koja omogućavanja pronalazak i kontinuirano iznajmljivanje studentskog smještaja. Poslovna logika aplikacije razvijena je u obliku REST API programskog sučelja uz pomoć Symfony razvojnog okvira, dok je korisničko sučelje implementirano uz pomoć NextJS biblioteke.

Aplikacija omogućava najmodavcima kreiranje i upravljanje oglasima za nekretnine, dok studentima omogućava detaljnu pretragu ali i stupanje u kontakt s najmodavcem uz pomoć ugrađenog komunikacijskog sustava. Osim toga aplikacija omogućava kreiranje i upravljanje korisničkim računom.

Ključne riječi: iznajmljivanje smještaja, NextJS, REST API, Symfony, web aplikacija

## **ABSTRACT**

In this paper, a web application has been developed that enables the search and continuous rental of student accommodation. The business logic of the application is developed in the form of a REST API interface using the Symfony framework, while the user interface is implemented using the NextJS library.

The application allows landlords to create and manage property listings, while students are enabled to perform detailed searches and contact the landlord using the built-in communication system. Additionally, the application allows for the creation and management of user accounts.

Keywords: accommodation rental, NextJS, REST API, Symfony, web application

## **PRILOZI**

PRILOG 1 – Datoteke i kod poslužiteljske aplikacije - <https://github.com/strikan95/cimi>

PRILOG 2 – Datoteke i kod klijentske aplikacije - <https://github.com/strikan95/Frontend-For-Cimi>

## **ŽIVOTOPIS**

Juraj Buljević rođen je 14.10.1995 u Osijeku. Pohađao je osnovnu školu „Vrbani“ u Zagrebu nakon koje završava srednju internacionalnu školu „Victor Hugo“ u Parizu. Akademske godine 2017/2018 upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, studij Elektrotehnika i informacijska tehnologija. Po završetku preddiplomskog studija upisuje diplomski sveučilišni studij Elektrotehnika, smjer komunikacije i informatika na istom fakultetu.

---

Potpis autora