

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

**Sveučilišni studij**

**RAZVOJ WEB APLIKACIJE ZA UPRAVLJANJE**  
**SMJEŠTAJNIM KAPACITETIMA**

**Završni rad**

**Luka Cvetković**

**Osijek, 2023.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 14.09.2023.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Luka Cvetković
<b>Studij, smjer:</b>	Programsko inženjerstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4477, 27.07.2020.
<b>OIB Pristupnika:</b>	69033139973
<b>Mentor:</b>	izv. prof. dr. sc. Josip Balen
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Razvoj web aplikacije za upravljanje smještajnim kapacitetima
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rada:</b>	U Završnom radu potrebno je izraditi web aplikaciju za upravljanje&nbsp;&nbsp;&nbsp;smještajnim kapacitetima za korisnike i administrator&nbsp;&nbsp;&nbsp;sustava.&nbsp;&nbsp;&nbsp;Administratori trebaju imati mogućnost upravljanja svojim objektima,&nbsp;&nbsp;&nbsp;gostima, zaposlenicima, računima, inventarom soba, itd. Korisnik ima mogućnost vidjeti sve objekte, dostupne sobe, rezervirati sobu, obaviti plaćanje.
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	14.09.2023.
<b>Datum potvrde ocjene od strane Odbora:</b>	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

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

Osijek, 25.09.2023.

Ime i prezime studenta:

Luka Cvetković

Studij:

Programsko inženjerstvo

Mat. br. studenta, godina upisa:

R4477, 27.07.2020.

Turnitin podudaranje [%]:

1

Ovom izjavom izjavljujem da je rad pod nazivom: **Razvoj web aplikacije za upravljanje smještajnim kapacitetima**

izrađen pod vodstvom mentora izv. prof. dr. sc. Josip Balen

i sumentora ,

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

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

Potpis studenta:

# SADRŽAJ

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak završnog rada</b> .....	2
<b>2. UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA</b> .....	3
<b>2.1. Koncept smještajnih kapaciteta</b> .....	3
<b>2.2. Upravljanje smještajem administrativne strane</b> .....	4
<b>2.3. Upravljanje rezervacijom smještaja korisničke strane</b> .....	4
<b>2.4. Postojeća rješenja problema upravljanja smještajnim kapacitetima</b> .....	5
<b>2.4.1. Booking.com</b> .....	6
<b>2.4.2. Airbnb</b> .....	7
<b>2.4.3. TripAdvisor</b> .....	9
<b>2.4.4. Vrbo</b> .....	10
<b>2.4.5. Expedia</b> .....	11
<b>3. TEHNOLOGIJE RAZVOJA WEB APLIKACIJE ZA UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA</b> .....	13
<b>3.1. Razvojno okruženje Visual Studio Code</b> .....	13
<b>3.2. REST API</b> .....	14
<b>3.3. Programski okvir Fastify</b> .....	14
<b>3.4. Upravljanje bazom podataka</b> .....	16
<b>3.5. Programski okvir Angular</b> .....	17
<b>4. OPIS RAZVOJA WEB APLIKACIJE ZA UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA</b> .....	19
<b>4.1. Razvoj poslužiteljske strane aplikacije</b> .....	19
<b>4.2. Razvoj klijentske strane aplikacije</b> .....	24
<b>5. ZAKLJUČAK</b> .....	32
<b>LITERATURA</b> .....	33

<b>SAŽETAK</b> .....	35
<b>ABSTRACT</b> .....	36
<b>ŽIVOTOPIS</b> .....	37
<b>PRILOZI</b> .....	38

# 1. UVOD

Prije digitalizacije i virtualizacije u turizmu, ljudi su rezervacije obavljali putem poziva ili dopisa. U današnjem digitalnom dobu upravljanje smještajnim kapacitetima postaje sve složenije zbog velike potražnje za različitim vrstama smještaja te potrebe za učinkovitim upravljanjem informacijama o rezervacijama i raspoloživosti. Stoga je bilo potrebno uvesti promjene u ovom području kako bi se procesi rezervacije i smještaja olakšali i ubrzali. Web aplikacije su se pojavile kao ključni alat u rješavanju ovog problema, omogućujući jednostavan pristup i učinkovito upravljanje smještajnim kapacitetima.

Izrada web aplikacije za upravljanje smještajnim kapacitetima zahtjeva poznavanje protokola i procesa potrebnih za rezervaciju smještaja digitalnim putem. Potrebno je poznavati i sintaksu programskih i označnih jezika (engl. *Markup Languages*) web aplikacija, što su u ovome slučaju HTML (engl. *HyperText Markup Language*), CSS (engl. *Cascading Style Sheets*) i TypeScript. Također je nužno znati koristiti biblioteke za klijentsku stranu razvoja aplikacije kako bi se ostvarilo korisničko sučelje koje je interaktivno i reaktivno. Uz to je potrebno razumjeti API-je (engl. *Application Programming Interface*) i znati primijeniti biblioteke poslužiteljske strane koji omogućuju komunikaciju s bazom podataka i logiku aplikacije [1].

Ova aplikacija nastoji riješiti različite probleme počevši sa kompliciranosti pretraživanja i rezerviranja smještaja jer klasični procesi rezervacije i pretraživanja zahtijevaju puno vremena te su često zamorni. Nadalje, ako bi se o nekom smještaju htjeli potpuno informirati, u dosta slučajeva bi morali posjetiti više web stranica ili nazvati više brojeva kako bi se informirali. Ova aplikacija nudi pristup svim tim informacijama na jednom mjestu za optimalno donošenje odluke o rezervaciji. Također, aplikacija omogućuje jednostavno sučelje za upravljanje rezervacijama i smještajnim prostorima koje vlasnicima smještaja omogućuje brz i efikasan pregled, izmjenu, ažuriranje i općenito upravljanje rezervacijama te izradu statističkih i analitičkih izvješća na temelju rezervacija. Korištenje ove aplikacije je dostupno svim korisnicima i vlasnicima 24 sata dnevno, sedam dana u tjednu, neovisno o geografskom položaju korisnika, dajući im pristup rezervacijama bilo kada i bilo gdje.

Ovaj rad se sastoji od tri glavna dijela. Prvi dio daje teorijsku podlogu o upravljanju smještajnim kapacitetima, kako izgleda proces rezervacije, te administrativna strana ovog posla i već postojeća rješenja koja se nalaze na internetu te približno rješavaju problem kao ova aplikacija. Drugi dio prikazuje opis tehnologija korištenih pri izradi te objašnjenje programskih okruženja korištenih za klijentsku i poslužiteljsku stranu aplikacije. U trećem i zadnjem dijelu aplikacije

prikazuju se koraci izrade logike određenih funkcionalnosti sa poslužiteljske strane, te interakcija sa korisnikom sa klijentske strane aplikacije.

### **1.1. Zadatak završnog rada**

Zadatak ovoga rada je izraditi web aplikaciju za upravljanje smještajnim kapacitetima za korisnike i administratore sustava. Administratori trebaju imati mogućnost upravljanja svojim objektima, gostima, računima, inventarom soba, itd. Korisnik ima mogućnost vidjeti sve objekte, dostupne sobe, rezervirati sobu, obaviti plaćanje, ostaviti recenziju i ocjene, upravljati svojom rezervacijom, itd. U radu je potrebno za razvoj front-enda koristiti HTML, CSS i Angular.js tehnologije dok za je za back-end potrebno koristiti Fastify (Node.js), a za upravljanje bazom podataka MySQL i slične tehnologije.

## **2. UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA**

U ovome se odlomku istražuju osnovne koncepti upravljanja smještajnim kapacitetima prouzročeni i tehnologije koje stoje iza tih sustava. Služi kao temelj za razvoj web-aplikacije koja je osmišljena kako bi unaprijedila upravljanje smještajnim kapacitetima.

### **2.1. Koncept smještajnih kapaciteta**

Smještajni kapaciteti su fizički resursi namijenjeni privremenom smještaju korisnika koji su u većini slučajeva turisti ili putnici. Ove mogućnosti uključuju različite vrste smještaja u koje spadaju hoteli, apartmani, moteli, kampovi, kuće, hosteli, itd. Pojam smještajnog kapaciteta uključuje sve prostore, jedinice ili sobe koje su omogućene za iznajmljivanje ili rezervaciju. Smještajni kapaciteti se sastoje od nekoliko elemenata, od kojih je najbitniji fizički prostor koji podrazumijeva prostore, jedinice ili sobe koje se koriste za smještaj gostiju. Mogu se razlikovati po svojoj veličini, estetskom izgledu, opremljenosti i namjeni. Fizički prostor može uključivati spavaće sobe, kupaoonice, dnevne sobe, kuhinje, terase i druge prostore. Nakon fizičkog prostora, najbitniji element je kapacitet, koji se odnosi na broj jedinica ili kreveta koji su dostupni korisnicima. Kapacitet se može mjeriti brojem kreveta ili brojem spavaćih soba. Na primjer, ako se gleda neki hotel može imati kapacitet od 200 soba, a apartmansko naselje može imati kapacitet od 20 apartmana.

Smještajni objekti mogu biti različitih tipova, stoga njihove mogućnosti mogu varirati ovisno o tipu ponuđenog smještaja. Na primjer, luksuzni hoteli s visokom razinom usluge i opremljenosti, obiteljske kuće za odmor s privatnim bazenima ili hosteli s više kreveta u zajedničkim sobama. Svaka vrsta smještaja ima svoje značajke i publiku za koju su predviđene. Smještajni objekti mogu nuditi dodatne pogodnosti i usluge kako bi privukli korisnike i pružili im bolje iskustvo. To može uključivati restorane, bazene, teretane, toplice, sobe za sastanke, usluge prijevoza, usluge čišćenja itd. Dodatni sadržaji i usluge često utječu na kvalitetu smještaja te mogu utjecati na cijene i konkurentnost. Ugostiteljske tvrtke trebaju sustav upravljanja rezervacijama za praćenje dostupnosti, prihvaćanje i potvrđivanje rezervacija te upravljanje dolascima i odlascima gostiju. Upravljanje rezervacijama podrazumijeva praćenje popunjenosti, postavljanje pravila otkazivanja smještaja, dodjelu objekata gostima, vođenje isprava o plaćanju i još mnogo toga. Upravljanje smještajnim objektima ima za cilj osigurati optimalno korištenje resursa, zadovoljstvo gostiju, rast prihoda i pravilan rad smještajnih objekata. Učinkovitim upravljanjem kapacitetima upravitelji usluga smještaja mogu pružiti kvalitetne usluge i zadovoljiti želje raznih skupina korisnika.



## **2.2. Upravljanje smještajem administrativne strane**

Upravljanje smještajnim kapacitetima podrazumijeva sve aktivnosti i protokole koji su usmjereni na efikasno korištenje, planiranje, organizaciju i kontroliranje smještajnih kapaciteta u turističkoj djelatnosti. Cilj svakog upravitelja takvim objektima treba biti pružanje optimalnog iskustva gostima i ostvariti uspješno poslovanje. Ono obuhvaća sljedeće stavke kao što su rezervacije i raspoloživost, smatrajući da kao administrator smještajnih kapaciteta, bitno je osigurati mogućnost rezervacije smještajnih kapaciteta prema potražnji. To podrazumijeva praćenje i ažuriranje dostupnosti smještajnih prostora, utvrđivanje pravila otkazivanja i osiguravanje učinkovitosti i pouzdanosti sustava rezerviranja.

Nadalje treba uzeti u obzir upravljanje cijenama smještaja koje ima važnu ulogu u postizanju optimalnih prihoda. To uključuje strategije za dinamičko prilagođavanje cijena na temelju sezonalnosti, potražnje, popunjenosti i drugih relevantnih čimbenika. Upravitelji smještaja također moraju pratiti i analizirati prihode, izvješćivati o financijama i identificirati prilike za povećanje prihoda. Kako bi posao upravljanja smještajem bio uspješan, upravitelj mora primjenjivati određene upravljačke aktivnosti. Upravljanje smještajnim objektima uključuje različite upravljačke aktivnosti kao što su čišćenje i održavanje prostora, upravljanje zalihama i opremom, koordinacija osoblja, osiguranje kvalitete usluge i usklađenosti s relevantnim propisima i standardima.

Još jedan bitan čimbenik za uspješno upravljanjem smještajnim kapacitetima je iskustvo korisnika. Postizanje visokog standarda iskustva korisnika vitalno je za uspješno upravljanje smještajnim prostorima. To uključuje brigu o gostima, pružanje personalizirane usluge, odgovaranje na povratne informacije i pritužbe te pružanje ugodne atmosfere koja zadovoljava potrebe i želje gostiju. Za kraj, bitno je da upravitelji smještajnih prostora razviju i znaju primijeniti marketinške taktike, što im omogućuje sposobnost privlačenja ciljane skupine gostiju. To uključuje izradu privlačnih marketinških materijala, rad s putničkim agencijama i internetskim platformama, upravljanje online prisutnošću i promocija jedinstvenih značajki i prednosti smještaja.

## **2.3. Upravljanje rezervacijom smještaja korisničke strane**

Problem upravljanja smještajem opisuje se u prethodnom poglavlju u ulozi upravitelja, odnosno administratora, a u ovome se poglavlju opisuje kako izgleda proces rezervacije smještaja sa strane korisnika. Prvi korak u procesu rezervacije za korisnika je pretraživanje različitih smještajnih opcija. To obično uključuje korištenje tražilica na internetu, web stranica za putovanja

ili aplikacija koje pružaju informacije o smještaju kao što su hoteli, apartmani, kuće za odmor itd. Korisnik može odrediti svoje želje kao što su lokacija, datum prijave i odjave, broj osoba i druge relevantne informacije. Nakon što je pretražio dostupne smještajne kapaciteta prema svojim željama, korisnik pregledava informacije o smještajima koji se nalaze u dobivenim rezultatima. To podrazumijeva informacije o sobi ili jedinici, fotografije, opise smještajnih objekata, ocjene i komentare drugih gostiju, cijene, pravila otkazivanja i druge informacije koje može smatrati bitnima. Korisnici mogu usporediti različite mogućnosti smještaja kako bi odabrali najprikladniju za svoje potrebe i mogućnosti.

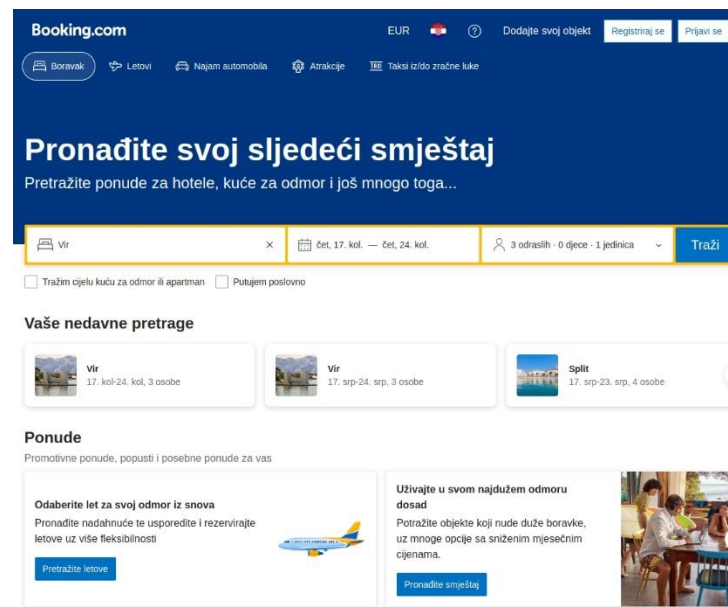
Kada je pregledao informacije o mogućnostima smještaja, korisnik odabire smještaj koji najviše i najbolje odgovara njegovima željama i potrebama. To može uključivati odabir određenih soba, vrsta kreveta ili drugih posebnih zahtjeva. Nakon odabira smještaja, korisnik započinje proces rezervacije unosom potrebnih podataka kao što su ime, kontakt podaci, datum prijave i odjave te broj gostiju. Osim toga, korisnik je obično dužan dati podatke o plaćanju za jamstvo rezervacije. Ovisno o platformi, korisnik može odabrati dodatne opcije kao što su doručak, prijevoz ili druge usluge. Nakon unosa svih potrebnih podataka, korisnik dobiva potvrdu rezervacije putem e-maila. Potvrda rezervacije sadrži detalje rezervacije, uključujući potvrdu datuma prijave i odjave gostiju, cijene, broj rezervacije, broj gostiju i druge relevantne informacije. U većini slučajeva od korisnika se traži plaćanje rezervacije. To može uključivati online plaćanje kreditnom karticom, PayPal-om ili drugim dostupnim načinima plaćanja. Nakon uspješne uplate, rezervacija je potvrđena. Poslije primljene potvrde, poželjno je putem kontakt broja telefona, ili mail adrese, kontaktirati vlasnika smještajnog objekta te sa njime dogovoriti pojedine detalje oko smještaja. Ti detalji mogu obuhvaćati dogovor o točnom vremenu useljenja, o određenim dijelovima ili postavkama smještaja koji su potrebni, odnosno nisu, te u slučaju da je omogućeno plaćanje smještaja prilikom dolaska na lokaciju, dogovor o mjestu i načinu plaćanja. Nakon toga ostaje samo da korisnik dođe na lokaciju smještaja, prilikom dolaska se registrira, te na odlasku odjavi.

#### **2.4. Postojeća rješenja problema upravljanja smještajnim kapacitetima**

U ovome poglavlju prikazuju se internetske platforme koje se bave problemom upravljanja smještajnim kapacitetima, isto kao i ova aplikacija. Opisuju se sljedeće aplikacije: Booking.com, Airbnb i TripAdvisor. Za svaku su dani primjere korištenja iz perspektive korisnika te kako izgleda pojedini dio aplikacije.

### 2.4.1. Booking.com

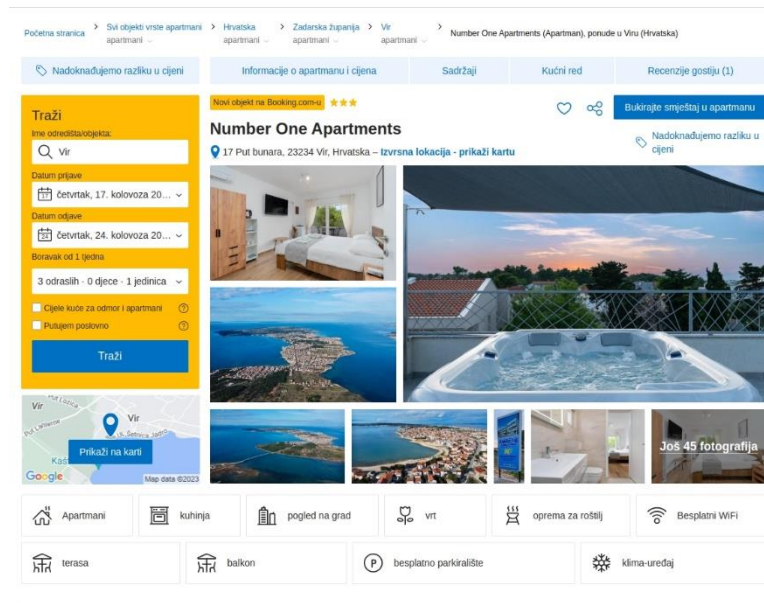
Booking.com je jedna od najpoznatijih i najvećih web aplikacija za rezervacije na svijetu. Uz široku ponudu hotela, apartmana, kuća i ostalih tipova smještajnih objekata, ona svojim korisnicima omogućuje i rezervaciju smještaja bilo gdje u svijetu. Slika 2.1. prikazuje naslovnu stranicu aplikacije koja nudi pretraživanje dostupnih smještajnih objekata prema lokaciji, datum boravka, te broju gostiju i jedinica. Ona također sadrži i povijest prethodnih pretraga, ponude za put do određene lokacije, pretragu po vrsti objekata, te prikazuje svoje prijedloge i najpopularnija odredišta u odabranoj državi. Klikom na gumb za pretraživanje, dobiva se lista rezultata po određenom mjestu, datumu i broju gostiju. Uz odabrane opcije pretrage smještaja, rezultate također možete izdvojiti prema vašem budžetu, ocjenama objekata, udaljenosti od centra odabranog mjesta, turističkim atrakcijama, vrstama soba, opremi objekata, itd. Lista rezultata se sastoji od objekata na kojima su prikazani: ime objekta, njihova cijena za odabrano vrijeme boravka, glavna slika jedinice, recenzije prošlih gostiju i ocjena udobnosti, te opće informacije kao što su oprema, odaljenost od centra (i plaže), dodatan opis i ostalo.



Slika 2.1. Naslovna stranica aplikacije Booking.com [2]

Klikom miša na određeni objekt, otvara se njegov profil. Slika 2.2. prikazuje profil koji se sastoji od naziva objekta, popisa svih njegovih slika, prikaza njegove točne lokacije na karti, opširnog opisa i ostalih bitnih informacija o objektu. Klikom na gumb za rezerviranje smještaja, kreće postupak rezervacije. Prvo se trebaju unijeti osnovni podaci, kao što su: ime, prezime, email adresa, broj telefona, te ukoliko korisnik ima određene zahtjeve, može ih napisati u rezervaciji, na

kraju korisnik unosi broj kartice kojom će platiti određeni smještaj i time završava proces rezervacije te se šalje potvrda o uspješnoj rezervaciji na navedeni e-mail.



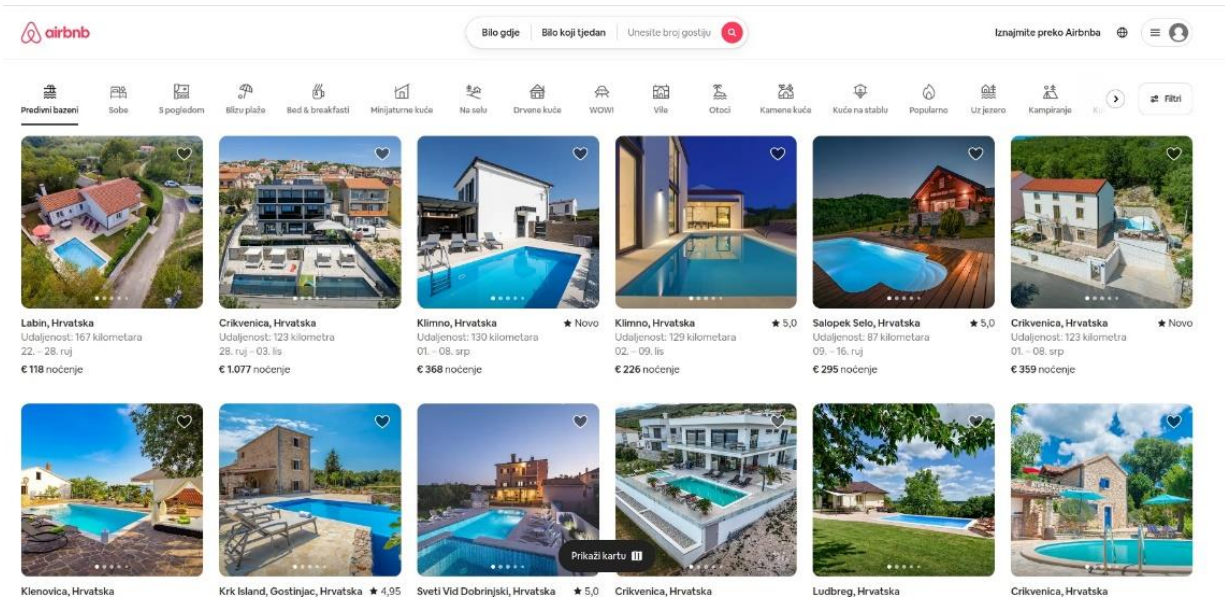
Slika 2.2. Profil smještajnog objekta u aplikaciji Booking.com [3]

## 2.4.2. Airbnb

Airbnb je popularna internetska platforma koja korisnicima omogućuje iznajmljivanje i rezervaciju raznih opcija smještaja, uključujući kuće, stanove, privatne sobe i pojedinačne nekretnine poput kuća za odmor ili brodova. Jedna od Airbnb-ovih ključnih značajki je mogućnost iznajmljivanja oglasa od domaćina koji posjeduju ili iznajmljuju oglase. Korisnici imaju mogućnost pregledavati različite profile objekata, pregledavati fotografije objekata, čitati recenzije gostiju i izravno komunicirati s upraviteljima prije donošenja odluke o rezervaciji smještajnog objekta. Airbnb pruža platformu koja povezuje putnike s domaćinima iz cijelog svijeta kako bi stvorili jedinstvena iskustva putovanja.

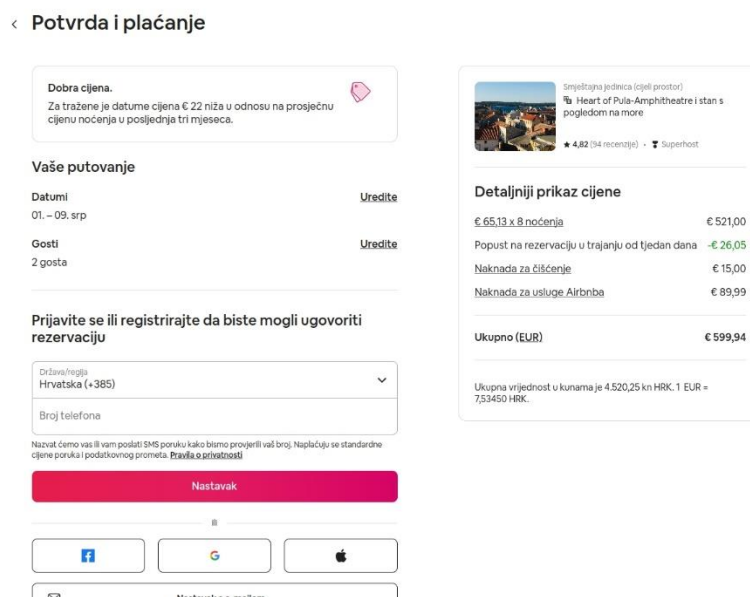
Slika 2.3.prikazuje izgled naslovnice Airbnb web aplikacije. Ona se sastoji od opcije za pretraživanje objekata po lokacije, vremenu boravka, te broju gostiju, slično kao i Booking.com. Uz to, početna stranica sadrži listu različitih tipova smještajnih objekata, te klikom na bilo koji od njih, otvara se popis najboljih, najpopularnijih i najpovoljnijih smještajnih objekata koji odgovaraju odabranom tipu objekta. Nadalje, kada korisnik pretraži objekte po određenim uvjetima, na ekranu se prikazuje popis svih objekata koji odgovaraju tim uvjetima. Uz nazive objekata, prikazani su kratki opis, recenzije korisnika, broj spavaćih soba, ukupna cijena noćenja, te lokacija svih objekata na karti. Također, uz primarne filtre, korisnik sadržaj može izdvojiti i

prema: rasponu cijena, vrsti smještaja, broju soba, sadržajima, značajkama pristupačnosti te jezicima koje domaćin govori.



Slika 2.3. Naslovna stranica aplikacije Airbnb [4]

Kada se korisnik odluči za željeni objekt, prije kretanja sa procesom rezervacije, on može komunicirati direktno sa domaćinom. Ovo je jedna od najkorisnijih značajki Airbnb-a jer tako korisnici upravitelju mogu postaviti određena pitanja o uslugama, sadržajima, pravilima smještaja i ostalim bitnim detaljima. Također, u nekim slučajevima gosti mogu imati posebne zahtjeve ili potrebe koje bi htjeli podijeliti s domaćinom. To mogu biti zahtjevi poput ranog dolaska ili ranog odlaska, dodatnih usluga, posebnih uvjeta ili drugih prilagodbi. Razmjena poruka omogućuje korisnicima komunikaciju o svojim željama izravno s domaćinom i pregovaranje o opcijama. Nakon što je sa upraviteljem dogovorio sve uvjete, korisnik kreće sa procesom rezervacije klikom na tipku za rezerviranje. Slika 2.4.prikazuje stranicu koja se otvori prilikom početka rezervacije. Tu korisnik unosi svoj broj telefona ili e-mail adresu uz pomoć koje će potvrditi svoju rezervaciju. Nakon toga, korisnika stranica vodi na odabir kartice kojom želi platiti svoj smještaj. Poslije unosa računa i obavljanja plaćanja, korisniku se isporučuje potvrda o uspješnoj rezervaciji i plaćanju, čime završava proces rezervacije.



Slika 2.4. Stranica za rezervaciju u aplikaciji Airbnb [5]

### 2.4.3. TripAdvisor

TripAdvisor je popularna internetska platforma za informacije o putovanjima i smještaju, te recenzijama i preporukama [6]. Uz informacije o turističkim odredištima i znamenitostima, TripAdvisor nudi i mogućnost pretraživanja i rezerviranja smještaja. Aplikacija prikazuje detaljne profile objekata, ocjene i komentare kupaca, fotografije, cijene smještaja i dostupnost. Korisnici mogu saznati više o iskustvima drugih putnika, ocijeniti smještaj i rezervirati ga putem povezanih aplikacija ili direktno putem TripAdvisor-a. TripAdvisor je postao jedan od poznatijih izbora za korisnike koji žele pristup sveobuhvatnim podacima i mišljenjima korisničke zajednice prije donošenja odluke o rezervaciji smještajnog objekta. Pretraživanje kreće odabirom lokacije, te nakon odabira, aplikacija prikazuje opis odabranog mjesta, fotografije, opis kulture, poznatih znamenitosti, vremenske uvjete te ostale bitne aspekte mjesta, što nam pokazuje Slika 2.5. Također daje savjete što raditi u tome mjestu, te koje lokacije posjetiti i kako provesti vrijeme. Uz to nudi i popis restorana i ostalih mjesta na kojima korisnici mogu objedovati. Za kraj nudi ono na čemu je korisnicima fokus, informacije o raznim mogućnostima smještaja na odredištu. Korisnici mogu pronaći opise hotela, recenzije gostiju, slike jedinica i smještaja, ponuđene usluge, te informacije o tome da li je smještaj dostupan. Podaci o smještaju pomažu korisnicima u donošenju odluke o tome koji smještaj najbolje zadovoljava njihove potrebe i želje.

## Explore Split



Europe > Croatia > Dalmatia > Split-Dalmatian County > Split

Split Tourism: Best of Split

### About Split

Thanks to its many former rulers (Roman, Austrian, Yugoslav, to name a few), Split is a European city unlike any other. What you get is a truly unique mix of architecture and history on every cobblestoned corner: ancient ruins, open-air flea markets, centuries-old cathedrals—it's a city seemingly frozen in time. And just beyond you'll find the Adriatic Sea, home to picturesque beaches and tranquil turquoise waters. Relax at Kašjuni or Bačvice Beaches or use it as a launching-

Slika 2.5. Informacije o odabranoj lokaciji u aplikaciji TripAdvisor [7]

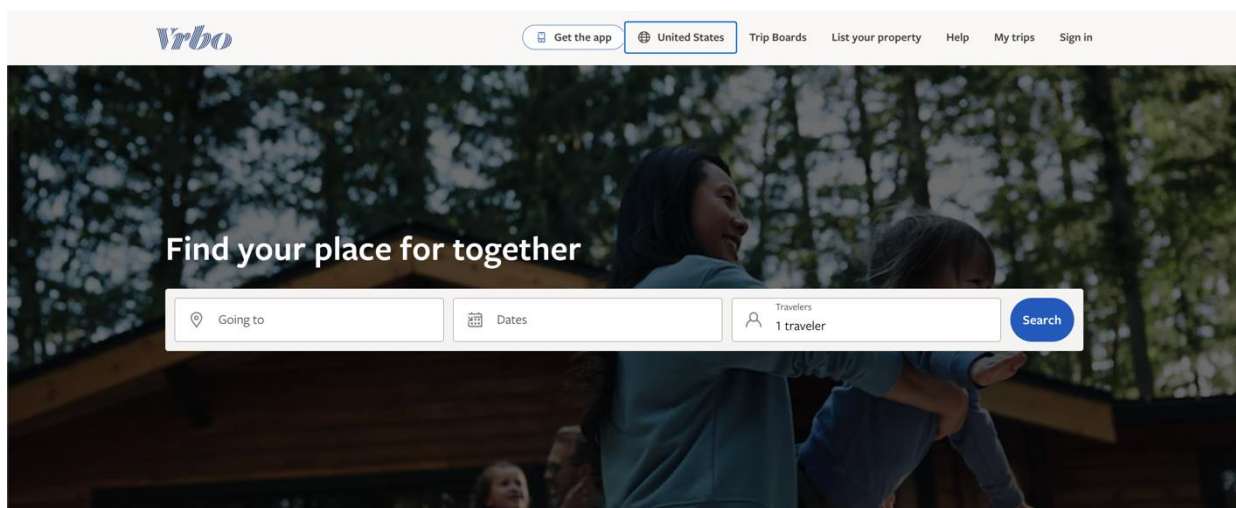
TripAdvisor omogućuje korisnicima da, nakon traženja smještaja i pregledavanja ocjena, budu usmjereni na određene web aplikacije preko kojih mogu dovršiti proces rezervacije njihovog smještaja. Ova aplikacija je uspostavila partnerstva sa drugim aplikacijama za rezervaciju, kao što su: Booking.com, Expedia, Hotels.com, itd. Oni omogućuju TripAdvisor-u da korisnicima ponudi pristup raznim opcijama smještaja koji se mogu rezervirati u ovim aplikacijama. Kada korisnik odabere poveznicu za rezervaciju smještaja na TripAdvisor-u, preusmjerava se na odabranu aplikaciju za rezervaciju. Ovaj korak daje korisniku mogućnost nastavka procesa rezerviranja na aplikaciji koju smatra provjerenom i prikladnom, te na kojoj ima vjerodostojnost. Veze za rezervaciju smještaja na TripAdvisor-u omogućuju korisnicima jednostavno prebacivanje između pregledavanja informacija o smještaju i procesa rezervacije. Osim toga, korisnici mogu pristupiti različitim opcijama smještaja i po različitim cijenama na različitim platformama za rezervaciju, što povećava fleksibilnost i pruža mogućnost pronalaženja najbolje ponude za korisnike.

#### 2.4.4. Vrbo

Vrbo je platforma usmjerena na iznajmljivanje odmorišnih kuća, apartmana, vikendica i drugih privatnih smještajnih objekata gdje vlasnici smještaja mogu oglašavati svoje objekte. Ona im daje mogućnost opisivanja svojih smještaja, postavljanja cijena, upravljanja dostupnošću i komuniciranja s potencijalnim gostima. Putnici mogu pregledavati ponude smještaja, pretraživati po destinacijama, datumima i broju gostiju, te rezervirati smještaj koji im odgovara. Vrbo pruža fotografije, recenzije i detaljne informacije o svakom smještaju. Također omogućuje komunikaciju između vlasnika smještaja i gostiju radi postavljanja pitanja ili dogovaranja detalja. Vrbo često

privlači obitelji i grupe putnika koji traže prostran i privatno smještaj za duža ili obiteljska putovanja.

Korisnik pristupa web stranici Vrbo i pregledava široku ponudu smještajnih objekata. Može pretraživati smještaj po odredištima, datumima putovanja i broju gostiju. Nakon što pronađe željeni smještaj, korisnik otvara stranicu smještaja kako bi vidio detalje, fotografije i cijene. Također može pročitati recenzije drugih gostiju. Korisnik odabire datume boravka i broj gostiju te pritisne gumb za rezervaciju. Ako je smještaj dostupan za odabrane datume, korisnik nastavlja na sljedeći korak. Korisnik unosi svoje osobne podatke i informacije o plaćanju kako bi potvrdio rezervaciju. Ovisno o politici smještaja, može biti potrebno platiti unaprijed ili putem drugih metoda. Nakon uspješne rezervacije, korisnik prima potvrdu putem e-pošte s detaljima rezervacije, kontaktom vlasnika smještaja i uputama za dolazak.



Slika 2.6. Naslovna stranica aplikacije Vrbo [8]

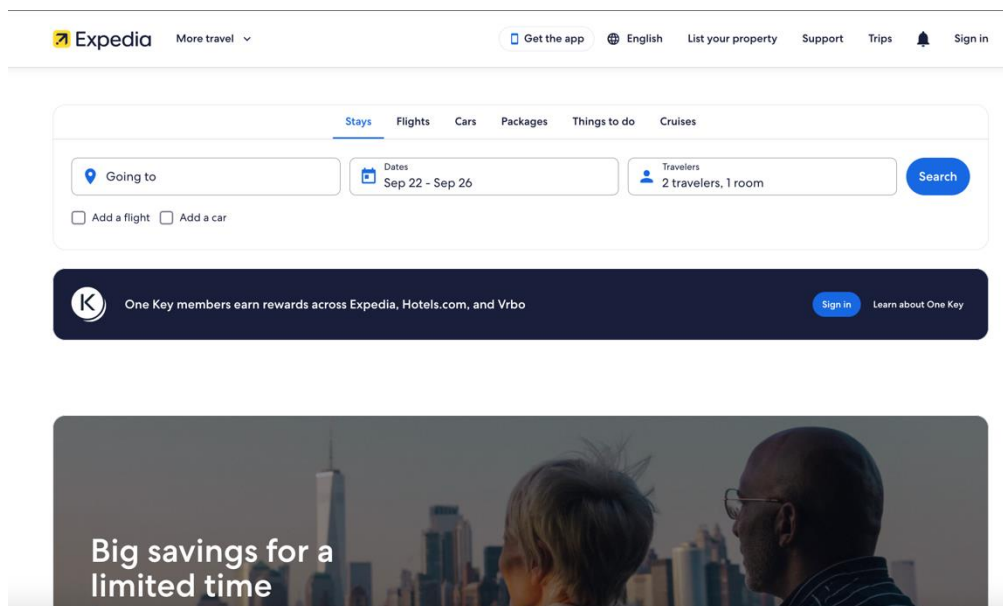
### 2.4.5. Expedia

Expedia je sveobuhvatna putnička platforma koja pruža širok spektar usluga, uključujući rezervaciju smještaja, letova, najma vozila i različitih aktivnosti. Putnici mogu koristiti Expedia-u za rezervaciju hotela, apartmana, kuća za odmor i drugih vrsta smještaja diljem svijeta. Ona također nudi mogućnost pakiranja različitih usluga kao što su letovi i smještaj u povoljnije cijene. Platforma nudi recenzije drugih putnika i korisne savjete za putovanje kako bi pomogla putnicima donijeti informirane odluke. Expedia privlači širok spektar putnika, uključujući pojedince, obitelji i poslovne putnike, zbog svoje raznolike ponude i opcija.

Proces rezervacije je vrlo sličan kao u prethodnim primjerima. Korisnik posjećuje web stranicu Expedia i započinje pretragu smještaja. Može pretraživati smještaj po odredištima,



datumima i broju gostiju. Korisnik pregledava različite ponude smještaja i otvara stranice smještaja kako bi vidio više informacija, fotografija i cijena. Nakon odabira smještaja, korisnik klikne na "Rezerviraj" i unosi svoje osobne podatke i informacije o plaćanju. Expedia omogućava plaćanje kreditnom karticom ili drugim dostupnim metodama. Nakon uspješne rezervacije, korisnik dobiva potvrdu putem e-pošte s detaljima rezervacije, kontaktom smještaja i uputama za dolazak.



Slika 2.7. Naslovna stranica aplikacije Expedia [9]

### 3. TEHNOLOGIJE RAZVOJA WEB APLIKACIJE ZA UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA

U ovome se poglavlju opisuju tehnologije potrebne za izradu web aplikacije koja obavlja poslove upravljanja smještajnim kapacitetima. Detaljno su opisana razvojna okruženja i programski okriivi te način na koji se primjenjuju prilikom izrade aplikacije.

#### 3.1. Razvojno okruženje Visual Studio Code

Za potrebe izrade ove aplikacije, koristi se VSC (engl. *Visual Studio Code*) kao uređivač teksta u kojemu je pisan programski kod [10]. VSC je jedan od najkorištenijih i najmoćnijih integriranih razvojnih okruženja (engl. *Integrated Development Environment*) koje korisnicima daje veliku količinu raznih značajki za razvoj web aplikacija. Jedan od najbitnijih razloga zbog kojeg je odabrano ovo razvojno okruženje je to što VSC nudi podršku za bogat skup programskih jezika i programskih okvira, a tu se ubrajaju jezici kao što su: JavaScript, TypeScript, HTML i CSS, te okviri poput Angular-a i Node.js-a. Nadalje, VSC je poznat po svome snažnom ekosustavu proširenja koji korisnicima omogućuju prilagođavanje svojeg okruženja kako bi najbolje odgovaralo njihovim potrebama. Korisnici mogu koristiti različita proširenja kako bi povećali svoju produktivnost, automatizirali zadatke, integrirali svoje aplikacije sa Git-om [11]. VSC ima ugrađenu podršku za Git koja podrazumijeva sljedeće glavne značajke: uvid u razlike u datoteci koja se uređuje u bočnom stupcu, Git traka stanja (engl. *Status Bar*) koji pokazuje trenutnu granu te dolazne i odlazne potvrde promjena (engl. *Commits*), inicijalizacija i kloniranje repozitorija, potvrda promjena, te sinkronizacija sa udaljenom granom i eventualno rješavanje sukoba pri spajanju [12]. VSC također uključuje napredne značajke uređivanja napisanog koda kao što su automatsko dovršavanje koda, preuređenje koda (engl. *Refactoring*), pretraživanje i zamjena teksta u kodu na razini projekta, gledanje izvornog koda u više prozora i slično. Ove značajke čine pisanje i održavanje koda lakšim i bržim. Korištenje VSC-a kao primarnog razvojnog alata daje mogućnost prilagodljivosti i podršku za učinkovit razvoj koda za izradu ove aplikacije.

Obzirom da se aplikacija izrađuje u operacijskom sustavu Ubuntu, VSC se instalira jednostavnim upisivanjem naredbe u naredbeni prozor (engl. *Terminal*) ``sudo apt install code``. Nakon što korisnik izvrši ovaj korak, VSC će biti uspješno instaliran na sustavu Linux. Način pokretanja VSC-a je da korisnik uđe u željenu mapu, te otvori naredbeni prozor u njoj i zatim u njega upiše ``code``. Na taj način će se VSC otvoriti u željenoj mapi, te u njemu korisniku daje mogućnost stvaranja novih mapa i tekstualnih datoteka u koje će upisivati svoj programski kod.

## 3.2. REST API

API, kako je ranije napisano u tekstu, definiran je kao skup metoda, protokola, te raznih alata koji daju korisniku mogućnost komunikacije među raznim programskim komponentama. API određuje pravila o tome kako će se komponente povezivati jedna sa drugom, razmjenjivati podatke i izvoditi potrebne operacije. API tipično pruža sučelje ili apstrakciju za postojeće programske komponente kako bi se lakše integrirali i upotrebljavali. On definira skup metoda i funkcija koje korisnicima omogućuju izvođenje određenih operacije, koje uključuju stvaranje, dohvaćanje, izmjenjivanje i brisanje podataka. Nadalje, API definira kako se podaci razmjenjuju između različitih komponenti. To podrazumijeva korištenje HTTP-a (engl. *Hypertext Transfer Protocol*), SOAP-a (engl. *Simple Object Access Protocol*), REST-a (engl. *REpresentational State Transfer*) ili bilo kojeg drugog komunikacijskog protokola koji definira pravila za stvaranje zahtjeva i odgovora.

U ovome se projektu koristi REST API kao protokol za razmjenu informacija među komponentama [13]. On je vrsta API-ja koja je bazirana na principima REST arhitekture. REST je vrsta dizajna kojoj je cilj poboljšati jednostavnost, skalabilnost i fleksibilnost u izgradnji web aplikacija. On se sastoji se neovisnih komponenata: poslužitelja, klijenata, međuspremnik, proxyja, međuspremnik za keširanje i tako dalje. Različiti ljudi su izradili ove komponente, te oni nikada nisu čuli jedni za druge prije nego što su počeli komunicirati, i mogu komunicirati samo tako da razmjenjuju dokumente preko HTTP-a (ili nekog sličnog protokola). Bitno je da se unaprijed dogovore o skupu protokolarnih semantika, inače komponente neće razumjeti jedna drugu [14]. REST sadrži resurse koji podrazumijevaju entitete koji obrađuje API. To su općenito objekti, zbirke, usluge ili bilo koji drugi logički entitet. Svaki resurs sadrži svoj jedinstveni identifikator (engl. *Uniform Resource Identifier*) koji se koristi za ciljanje i pristup određenom resursu putem API-ja, te se može predstaviti u raznim formatima, što su najčešće JSON (engl. *JavaScript Object Notation*) ili XML (engl. *eXtensible Markup Language*) format. REST API koristi standardne HTTP metode kao što su GET, POST, PUT i DELETE kako bi definirali operacije resursa. On također ima svojstvo bezstanovnosti, što znači da svaki zahtjev mora sadržavati sve informacije koje su potrebne kako bi se zahtjev obradio. Sve informacije o statusu moraju biti uključene u sam zahtjev, a ne u sam API.

## 3.3. Programski okvir Fastify

U ovome projektu se koristi programski okvir Fastify za stvaranje poslužiteljskog dijela aplikacije [15]. Fastify je moderan, izrazito brz i učinkovit programski okvir za Node.js. Razvijen je za omogućavanje stvaranja učinkovitih API-ja i web aplikacija. On je popularan po svojoj

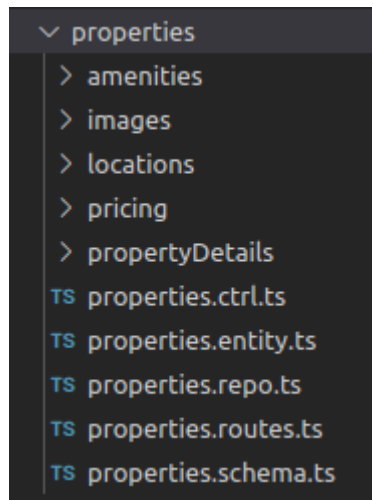
jednostavnosti kada je u pitanju korištenje, skalabilnost i zauzimanje memorije. Instalira se upisivanjem naredbe `npm install fastify` u naredbeni prozor. Jedna od najbitnijih značajki Fastify-a zbog kojih je odabran za izradu ove aplikacije jest njegova iznimna brzina, koju postiže uporabom asinkronih operacija, minimalnih ovisnosti i pojednostavljenom obradom HTTP zahtjeva. On je napravljen za učinkovito korištenje resursa, što znači da troši minimalne procesorske i memorijske resurse, što mu omogućuje obradu velikog broja zahtjeva uz minimalan učinak na performanse sustava. Također, prilikom odabira programskog okvira za poslužiteljsku stranu, važan je čimbenik to što se tehnologija Fastify sve više koristi te postoji mogućnost da će postati glavni programski okvir za NodeJs, što je trenutno Express.js, koji je na tržištu već duži niz godina, dok je Fastify relativno nov okvir.

Kako bi se programski kod optimizirao i pojednostavio, koristi se obrazac za dizajn koji se naziva Resursni obrazac za upravljanje podacima (engl. *Resource-oriented design*) [16]. On se često koristi u razvoju REST API-ja, jer podržava transparentnu organizaciju resursa, upotrebljivost HTTP protokola i skalabilnost. Ovaj pristup znatno olakšava izgradnju trajnih i sveobuhvatnih sustava koje je lako razumjeti i održavati. Ovaj obrazac se sastoji od resursa od kojih svaki ima:

- Entitet – predstavlja podatkovni model koji ima svoje atribute i relacije na druge povezane tablice, u čijem obliku će se kreirati tablice u bazi podataka te ih dohvaćati od tamo.
- Repozitorij – sloj koji dolazi direktno nakon entiteta, on komunicira sa bazom podataka te pruža metode ta dohvaćanje, spremanje, izmjenu i brisanje podataka u bazi. Repozitorij skriva pojedinosti podatkovne komunikacije i pruža apstrakciju za ostatak aplikacije.
- Kontroler – zadužen je za obradu HTTP zahtjeva sa strane klijenta. On prihvata zahtjeve, poduzima potrebne radnje i kreira odgovore. Kontroler često koristi usluge i sloj repozitorija za dohvaćanje ili manipulaciju podacima i odgovaranje korisniku rezultatima ili pogreškama.
- Shema – skup definicija koje opisuju strukturu i provjeru valjanosti podataka. Shema se može definirati korištenjem objektno-relacijskog mapiranja ili na druge načine. Ona određuje oblik i ograničava podatke pohranjene u bazi.
- Rute – predstavljaju krajnje točke kroz koje su dostupne funkcije izložene putem API-ja. Svaka ruta je povezana sa određenim kontrolerom i specifičnim HTTP metodama. One određuju URL-ove, zahtjeve koji se postavljaju na određenoj ruti i radnje koje se poduzimaju.

- Podresursi – predstavljaju entitete koji su uglavnom povezani samo na svoj roditeljski entitet te stoga ovise o njemu i nemaju svoje određene funkcionalnosti. Zbog njegovih karakteristika, spremamo ga unutar roditeljskog resurse.

Slika 3.1.prikazuje primjer korištenja resursnog obrasca u mojoj aplikaciji. “properites” predstavlja glavni resurs, koji ima svoj kontroler, entitet, repozitorij, rute i shemu, dok su “amenities”, “images”, “locations”, “pricing” i “propertyDetails” podresursi.



Slika 3.1. Prikaz strukture resursnog obrasca u web aplikaciji

### 3.4. Upravljanje bazom podataka

Kako bi poslužiteljskom dijelu bio omogućen pristup bazi podataka, potrebno je koristiti objektno-relacijsko mapiranje iliti ORM (engl. *Object–relational mapping*), te se za obavljanje ovih poslova u ovoj aplikaciji koristi alat TypeORM [17]. To je biblioteka objektno-relacijskog mapiranja za jezike TypeScript i JavaScript koja pojednostavljuje obavljanje posla s relacijskim bazama podataka. Programerima omogućuje korištenje objekata i klasa za komunikaciju s bazom umjesto direktnog pisanja SQL (engl. *Structured query language*) upita. TypeORM podržava razne baze podataka kao što su MySQL, PostgreSQL, SQLite i slične. On nudi mogućnosti upotrebe mapiranja objekata u tablice baze podataka, kontrolu relacija među povezanim tablicama, kreaciju SQL upita, migracije i transakcije baze podataka.

Za upravljanje bazom podataka koristi se TypeORM i MySQL [18]. Prvo je potrebno instalirati MySQL poslužitelj i TypeORM na osobno računalo. Nakon toga, potrebno je uspostaviti vezu s bazom podataka u TypeORM konfiguracijskoj datoteci, gdje je potrebno navesti podatke poput: hosta, porta, korisničkog imena, zaporke i naziva baze podataka. Poslije toga se uz pomoć TypeORM-a kreiraju entiteti koji predstavljaju tablice u MySQL bazi podataka, te se po potrebi prave migracije i pišu određeni SQL upiti. Nakon definiranja entiteta, migracija i upita, koristi se

TypeORM za sučelje sa web aplikacijom. To uključuje postavljanje i inicijaliziranje aplikacije te upotrebu entiteta i upita za pristup podacima iz MySQL baze podataka.

SQL jezik je podijeljen u nekoliko različitih dijelova:

- SQL naredbe za shemu, koje se koriste za definiranje struktura podataka pohranjenih u bazi podataka
- SQL naredbe za podatke, koje se koriste za manipulaciju podatkovnim strukturama koje su prethodno definirane pomoću SQL naredbi za shemu i
- SQL naredbe za transakcije, koje se koriste za početak, završetak i poništavanje transakcija [19].

### **3.5. Programski okvir Angular**

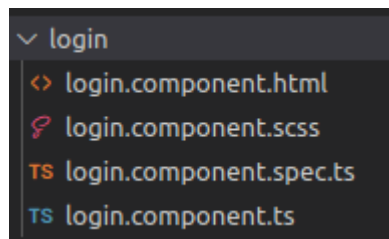
Angular je jedan od najpoznatijih i najmoćnijih programskih okvira za razvoj i izradu web aplikacija koji daje korisnicima mogućnost izgradnje modernih, skalabilnih i brzih aplikacija [20]. Upotreba Angulara kao okvira za ovaj rad je učinkovita jer se oboje temelje na JavaScript i TypeScript jezicima, što olakšava implementaciju i povezivanje klijentske i poslužiteljske strane aplikacije.

Nebular je skup komponenti korisničkog sučelja (UI) izgrađenih na temelju Angulara [21]. Nudi bogat skup gotovih komponenti i alata za olakšavanje izrade profesionalnih i suvremenih aplikacija. Prilikom instalacije Nebulara, kreira se skup komponenti koje je moguće koristiti u klijentskom dijelu aplikacije, a neke od najbitnijih komponenti su:

- Navigacija – omogućava izgradnju sustava kao što su izbornici, bočne trake, kartice i oznake. Ona služi kako bi se korisnik lakše organizirao i navigirao po sadržaju aplikacije.
- Obrasci – za stvaranje obrazaca, Nebular nudi mogućnosti unosa teksta, potvrdnih okvira, switcheva, padajućih izbornika i slično. One imaju ugrađenu provjeru valjanosti i omogućuju jednostavan unos podataka od korisnika
- Tablice i popisi – omogućuje tablične podatke, označavanje stranica, filtriranje i sortiranje. One pomažu organizirati i prikazati podatke na uredan način
- Modali i dijalozi – kako bi se omogućila interakcija korisnika s modalnim prozorima i dijalogima, omogućene su komponente koje olakšavaju procese otvaranja, zatvaranja i prikaza dodatnih informacija na stranici
- Grafikoni – koriste se za vizualizaciju podataka, u Nebularu je moguće implementirati različite vrste grafikona, kao što su linijski, stupčasti, kružni, itd.

Nebular, uz prethodno objašnjene primjere, nudi niz alata, tema i stilova koji korisniku pomažu u stvaranju atraktivnih i dosljednih korisničkih sučelja. Ovaj skup komponenti omogućuje da se aplikacija razvija brže i efektivnije jer pomaže korisniku da ne piše složen kod od nule. Slika 3.2. prikazuje dizajn Nebular komponenti koji slijedi pravila i obrasce određene u Angularu, on se sastoji od:

- HTML predloška - podrazumijeva sučelje komponente prema korisniku i određuje njegovo prikazivanje na stranici. U njemu se određuju gumbi, obrasci, tablice, kartice i slično. Model koristi odgovarajuće putanje i sintaksu Angulara ta prikaz i komunikaciju sa korisnikom,
- TypeScript klase - označava kreiranje logike komponente. U njoj se definiraju metoda, svojstva i ponašanje, te se obrađuju korisnički događaji, manipulacija sa informacijama, komunikacija usluge i izvođenje drugih operaciju koje odgovaraju komponenti,
- CSS stila – primjenjuje elemente korisničkog sučelja kako si se pružila dodatna mogućnost ili promjena njihovog izgleda i ponašanja.



Slika 3.2. Struktura obrasca Nebulara u web aplikaciji

## 4. OPIS RAZVOJA WEB APLIKACIJE ZA UPRAVLJANJE SMJEŠTAJNIM KAPACITETIMA

U ovome se poglavlju opisuje razvoj i izrada web aplikacije koja obavlja poslove upravljanja smještajnim kapacitetima, rezervacijama, objektima i korisnicima sa administrativne strane. Nakon toga se prikazuju poslovi rezervacije, kreacije profila objekata, dodavanje recenzija, obavljanje procesa plaćanja i ostali poslovi koji spadaju u proces upravljanja smještajnim kapacitetima sa korisničke strane.

### 4.1. Razvoj poslužiteljske strane aplikacije

Razvoj aplikacije sa poslužiteljske strane kreće instalacijom potrebnih ovisnosti za Fastify i TypeORM i kreacijom osnovne datoteke za poslužitelja pod nazivom `index.ts`. U toj se datoteci nalaze funkcije za registraciju svih potrebnih ovisnosti te postavljanje domaćina (engl. *host*) i luke (engl. *port*) na kojemu se poslužitelj nalazi na internetskoj mreži. U njoj se također podešava i inicijalizira komunikacija sa bazom podataka putem TypeORM-a koji definira podatke potrebne za spajanje na bazu, kao što su: tip baze podataka, domaćin, luka, korisničko ime, zaporka, ime baze podataka, te definiranje tablica sa podacima u bazi. Nakon što je ovo sve učinjeno, kreće se na definiranje potrebnih resursa koji se koriste za aplikacijsko sučelje. Kako je ranije objašnjeno, u resursu se prvo kreira entitet koji predstavlja tablicu u bazi podataka. Ova je aplikacija podijeljena na pet resursa, od kojih svaki ima svoju ulogu i funkcionalnost u razvoju aplikacije.

Korisnik može pristupiti aplikaciji kao administrator (engl. *Admin*) i kao standardni korisnik (engl. *User*). Proces kreiranja resursa kreće izradom resursa *Admin*. Aplikacija mu omogućuje pristup i upravljanje svim smještajima, rezervacijama, recenzijama te korisnicima u aplikaciji. Također ima pristup analitičkim podacima o smještajnim kapacitetima, kao što su: ukupan broj smještaja, koliko ih ima po kategorijama (apartmani, hoteli, vile), te o svakom smještajnom objektu pojedinačno: prosječna ocjena, ukupni prihod i koliko objekt ima rezervacija. Njegova se tablica u bazi podataka sastoji od sljedećih atributa:

- ID – jedinstveni identifikacijski broj administratora
- Korisničko ime (engl. *Username*) - korisničko ime prijavljenog administratora
- Email – email adresa kojom se administrator prijavio u aplikaciju
- Zaporka (engl. *Password*) – zaporka kojom se prijavio u sustav
- Ime (engl. *First name*)
- Prezime (engl. *Last name*)



- Broj telefona (engl. *Phone number*)

Uz gore navedeno, administrator ima i metodu kojom se, prilikom unosa podataka, zaporka kriptira uz pomoć alata Bcrypt [22]. Nakon toga, ona se sprema u bazu podataka kao kriptirani skup znakova kako bi se osigurala sigurnost pristupa profilu kako prikazuje Slika 4.1. Također ima i funkciju koja prilikom unosa zaporke, tijekom procesa prijave, provjerava je li unesena zaporka odgovara onoj u bazi koja je kriptirana. Programski kod 4.1.prikazuje rutu uz pomoć koje se administrator prijavljuje u aplikaciju. Ona sadrži potrebne provjere uz koje se proces prijave odvija čisto i bez ikakvih propuštanja, te ako jedan od zahtjeva nije ispunjen, poslužitelj korisniku javlja pogrešku.

id	username	password	email
1	marin_maric	\$2a\$10\$amNTmd6tCLtjLnN8axq2mugOHOegsmeTdiqsMfFn3GxI	marin@mail.com

Slika 4.1. Podaci o korisniku u bazi podataka

```
fastify.post(
  "/api/admin/login",
  { schema: adminLoginSchema },
  async (req, reply) => {
    const admin = await adminRepo.findOne({ where: { email: req.body.email } });
    if(!admin) {
      return reply.code(400).send({ message: "Admin not found!" });
    }
    if(!req.body.password) {
      return reply.code(400).send({ message: "Password wasn't sent!" });
    }
    const checkPassword = await admin.comparePassword(req.body.password);
    if(!checkPassword) {
      return reply.code(400).send({ message: "Invalid password!" });
    }
    else {
      const token = await fastify.jwt.sign({ id: admin.id, role: "Admin" })
      return reply.send({ token: token })
    }
  }
);
```

Programski kod 4.1. Ruta za prijavu u aplikaciju kao administrator

Nadalje, drugi način prijave korisnika aplikacije je *User*, koji također predstavlja i resurs u poslužiteljskom dijelu. Ovaj resurs daje korisniku mogućnost dohvaćanja, uređivanja i brisanja svojih podataka. Također, omogućuje mu upravljanje omiljenim smještajnim objektima, te plaćanje svoje rezervacije smještaja čiji je vlasnik drugi korisnik aplikacije. Uz to, sadrži dva podresursa: obavijesti i kreditne kartice. Podresurs obavijest predstavlja obavijesti koje korisnik

prima od administratora i drugih korisnika. One se mogu ticati njegovog smještajnog objekta, rezervacije ili recenzija koje je ostavio na drugim objektima. Podresurs kreditna kartica predstavlja sredstvo plaćanja korisnika kojima on podmiruje troškove boravka u smještajnim kapacitetima. On se sadrži od atributa specifičnih za kreditne kratice, kao što su: broj računa, ime vlasnika, mjesec i godina do koje je kartica valjana i ccv (engl. *Card Code Verification*). Kako sadrži podresurse, resurs korisnik omogućuje korisniku i upravljanje kreditnim karticama te obavijestima. Ovaj se entitet sastoji od gotovo istih atributa kao i administrator, samo što uz osobne informacije i metode za kriptiranje šifre, sadrži i sljedeće relacije:

- Smještajni objekti (engl. *Properties*) – svi smještajni objekti koji su u vlasništvu korisnika
- Omiljeni objekti (engl. *Favorites*) – smještajni objekti koje je korisnik označio kao omiljene
- Rezervacije (engl. *Bookings*) – sve korisnikove rezervacije
- Obavijesti (engl. *Notifications*)
- Kreditne kartice (engl. *Credit cards*)

Sljedeći i najbitniji resurs jest *Property*. On predstavlja smještajne objekte koje dodaju korisnici te ih zatim drugi korisnici mogu rezervirati. Sadrži se od osnovnih atributa kao što su vlasnik, ime objekta, opis i minijatura, te kako je ovaj resurs relativno kompleksan, podijeljen je na više podresursa:

- Detalji (engl. *Property details*) – detalji kao što su tip objekta, tip sobe, maksimalan broj djece i odraslih koji smiju boraviti u objektu, broj kupaonica, spavaćih soba i kreveta
- Određivanje cijena (engl. *Pricing*) – informacije o cijeni po noćenju, minimalan ostanak te eventualan popust
- Lokacija (engl. *Location*) – sadrži državu, županiju, poštanski broj, grad, točnu adresu te latitudu i longitudu objekta
- Sadržaji (engl. *Amenities*) – dodatne pogodnosti koje nudi smještaj, kao što su: parking, roštilj, teretana, bazen, balkon

Ovaj resurs omogućuje korisniku dodavanje novog smještajnog objekta te dohvaćanje i upravljanje svim svojim objektima. Uz to, omogućuje mu dohvaćanje svih dostupnih objekata u aplikaciji kako bi odabrao koji je pogodan za njegovu rezervaciju. Također može i prilagoditi postavke po kojima želi izdvojiti objekte koji ga zanimaju, a te postavke su lokacija, datum prijave i odjave te broj gostiju, što je odrađeno u funkciji koju pokazuje Programski kod 4.2.U toj se funkciji koriste SQL upiti za komunikaciju sa bazom podataka kako bi se dohvatili samo oni

objekti koji zadovoljavaju uvjete u `where` i `andWhere` funkcijama. Kada korisnik otvori detaljan profil jednog objekta, on uz pomoć odgovarajuće rute sa poslužitelja može provjeriti da li se njegov izabrani datum prijave i odjave preklapa sa nekom drugom rezervacijom istog objekta i da li može ostaviti recenziju, ovisno o tome je li prethodno posjetio odabrani objekt.

Resurs u kojemu je implementirano rezerviranje smještajnih objekata naziva se *Booking*. Entitet u obliku kojega se podaci o rezervaciji spremaju u bazu podataka, sastoji se od sljedećih atributa: korisnika koji je rezervirao smještaj, objekta koji je rezerviran, datuma prijave i odjave, broja djece i odraslih osoba koji odsjedaju u objektu, broja noćenja, te atributa tipa `boolean`, što znači da može poprimiti vrijednosti točno ili netočno, a on služe za provjeru je li rezervacija plaćena. Rute u ovome resursu omogućuju korisniku rezervaciju smještaja, te dohvaćanje svih svojih rezervacija i u slučaju da nije u mogućnosti odsjesti u objektu u rezerviranom vremenskom periodu, može urediti svoju rezervaciju ili ju otkazati.

```
async searchProperties(query) {
  return await _pR
    .createQueryBuilder('property')
    .leftJoinAndSelect('property.amenities', 'amenities')
    .leftJoinAndSelect('property.images', 'images')
    .leftJoinAndSelect('property.pricing', 'pricing')
    .leftJoinAndSelect('property.bookings', 'booking')
    .leftJoinAndSelect('property.location', 'location')
    .leftJoinAndSelect('property.propertyDetails', 'propertyDetails')
    .where('(booking.id IS NULL OR (booking.check_out_date <= :check_in_date OR
booking.check_in_date >= :check_out_date))', {
      check_in_date: query.check_in_date,
      check_out_date: query.check_out_date,
    })
    .andWhere('location.city = :city', { city: query.location })
    .andWhere('propertyDetails.max_adults >= :adults AND propertyDetails.max_kids >=
:kids', {
      adults: query.adults,
      kids: query.kids,
    })
    .getMany();
}
```

**Programski kod 4.2.** Funkcija u repozitoriju smještajnih objekata za pretraživanje po određenom kriteriju

Programski kod 4.3.prikazuje funkciju u repozitoriju koja obrađuje unesene podatke, te radi provjere kako bi se vidjelo je li moguće rezervirati objekt sa predanim podacima. Ako je moguće, šalje se obavijest vlasniku objekta da je primio novu rezervaciju od prijavljenog korisnika.

Posljednji resurs koji se koristi za izgradnju poslužiteljskog dijela aplikacije je *Review*. On predstavlja recenzije koje su korisnici ostavili nakon odsjedanja u određenom smještajnom objektu. Tablica u bazi podataka se sastoji od korisnika koji je ostavio recenziju, objekta na koji se recenzija odnosi, ocjene od jedan do pet, te naslova i komentara recenzije. Rute korisniku omogućavaju ostavljanje recenzije na objekt koji je posjetio, uređivanje ili brisanje svoje recenzije, te dohvaćanje svih recenzija koje su ostavili na smještajnim objektima.

Kako je aplikacija podijeljena na dva tipa korisnika, administratora i klasičnog korisnika, potrebno je odrediti koje rute su namijenjene samo za administratora, a koje za korisnika. Za rješenje ovoga problema se koristi JSON token za autentikaciju (engl. *JSON Web Token*), ili skraćeno JWT [23]. To je lagan, siguran i samostalan način za razmjenu podataka između dvije strane u obliku JSON objekata. Uglavnom se koristi kako bi se korisnik autenticirao i autorizirao u web aplikaciji, uključujući poslužiteljske sustave.

```
async bookProperty(userId: number, data) {
  const user = await _uR.findOne({ where: { id: userId }});
  const property = await _pR.findOne({ relations: { pricing: true, propertyDetails: true,
bookings: true, owner: true }, where: { id: data.propertyId }});
  const overlappingBookings = property.bookings.filter((booking) => (booking.check_in_date
<= data.check_in_date && booking.check_out_date >= data.check_in_date) || (booking.check_in_date
<= data.check_out_date && booking.check_out_date >= data.check_out_date));
  if (overlappingBookings.length > 0) { throw new Error("Dates already taken!") }
  if(data.adults_number > property.propertyDetails.max_adults || data.kids_number >
property.propertyDetails.max_kids) { throw new Error("Too many people included in the booking!")}
  let total_price = property.pricing.price_per_night * data.number_of_stays;
  if(property.pricing.discount_percent !== 0) {
    total_price = total_price - total_price * (property.pricing.discount_percent / 100);
  }
  await _nR.save(_nR.create({
    user: property.owner,
    title: "Property booking",
    content: `Your property ${property.title} has been booked by user ${user.first_name}
${user.last_name}.`
  }));
  return await _bR.save(_bR.create({ user: user, property: property, check_in_date:
data.check_in_date, check_out_date: data.check_out_date, adults_number: data.adults_number,
kids_number: data.kids_number, total_price: total_price, number_of_stays: data.number_of_stays
}));
}
```

**Programski kod 4.3.** Funkcija u repozitoriju rezervacija za izradu nove rezervacije

Za korištenje JWT alata, u aplikaciji se stvara dodatak za Fastify koji omogućuje njegovu implementaciju u sustav. Programski kod 4.4.prikazuje dekorater za Fastify koji se naziva ``adminACL`` i u njemu se nalazi funkcija ``jwtVerify`` uz pomoć koje se provjerava JWT token u

primljenom zahtjevu. Ako je token nevažeći, nedostaje ili je uloga korisnika u tokenu `User`, a ne `Admin`, vraća se odgovor koji govori korisniku da ne može pozvati rutu jer nije pravilno autoriziran. Kako za administratora, tako postoji i dekorater za korisnika, a oni se predaju rutama u atribut `preValidation` koji označava da korisnik mora biti prijavljen na određeni način kako bi pristupio ruti, što pokazuje Programski kod 4.5.

```
fastify.decorate("adminACL", async (req, reply) => {
  fastify.aclFactory
  {
    const token = await req.jwtVerify();
    if (token.role === "Admin") { }
    else {
      reply.send("Unauthorized");
    }
  }
})
```

**Programski kod 4.4.** Dekorater za provjeru JWT tokena administratora

```
fastify.route({
  method: "GET",
  url: "/api/users",
  preValidation: fastify.userACL,
  handler: (await userController).getMe,
  schema: getMeSchema
});
```

**Programski kod 4.5.** Ruta koja zahtjeva da korisnik bude prijavljen kako bi ju izvršio

## 4.2. Razvoj klijentske strane aplikacije

Nakon izrade poslužiteljskog dijela, potrebno je izraditi klijentski dio aplikacije. Ona kreće inicijalizacijom `ApiService` klase koja je zadužena za uspostavljanje komunikacije između poslužiteljske i klijentske strane aplikacije. U to se ubraja slanje zahtjeva kako bi se dohvatili podaci, zatim slanje podataka na poslužitelja, izmjenu te brisanje podataka o smještajnim objektima, korisnicima i rezervacijama. Poslije inicijalizacije, kreiraju se metode kojima se omogućava komunikacija s API-jem. Tu su uključene metode `GET`, `POST`, `PUT` i `DELETE`, koje odgovaraju određenim vrstama zahtjeva. Te se metode koriste u komponentama kako bi obradili podatke iz baze. Programski kod 4.6.prikazuje kreaciju varijable zaglavlja u kojoj se iz lokalnog skladišta dohvaća JWT token trenutno prijavljenog korisnika te se on koristi kako bi se mogle pozivati rute iz poslužitelja koje zahtjevaju autentikaciju. Nadalje se uspostavlja povezivanje sa poslužiteljem na kojemu je pokrenut API. Metoda `registerUser` poziva rutu za

registraciju iz API-ja te mu se predaje tijelo koje sadrži podatke o kreiranom korisniku koje API obrađuje, te ukoliko je registracija uspješna, vraća JWT token i objekt korisnika sa njegovim atributima.

```
export class ApiService {
  private _headers = new HttpHeaders({
    "Content-Type": "application/json",
    Authorization: "Bearer " + localStorage.getItem("token"), "Access-Control-Allow-Origin": "*",
  });
  private _base_api = "http://localhost:3100/"
  constructor(private _http: HttpClient) {
  }
  registerUser(body) {
    return this._http.post<any>(
      `http://localhost:3100/api/users/register`, body,
    )
  }
}
```

**Programski kod 4.6.** Servis za povezivanje klijenta i poslužitelja

Nakon implementacije `ApiServicea`, potrebno je definirati stranice i komponente koje su bitne za realizaciju korisničkog sučelja aplikacije. Kako je rečeno u dijelu za razvoj poslužiteljske strane, u aplikaciju se moguće prijaviti kao administrator i standardni korisnik. Slika 4.2. prikazuje `login-register` komponentu koja korisniku omogućava stvaranje računa u aplikaciji, te prijavu kao administrator ili kao korisnik. Prilikom obavljanja ove radnje, u lokalno skladište se sprema JWT token korisnika ili administratora koji mu omogućuje da u daljnjim komponentama upravlja aplikacijom.

Sign in'."/>

WANDER BUDDY

### Register now

First name:

Last name:

E-mail:

Username:

Phone:

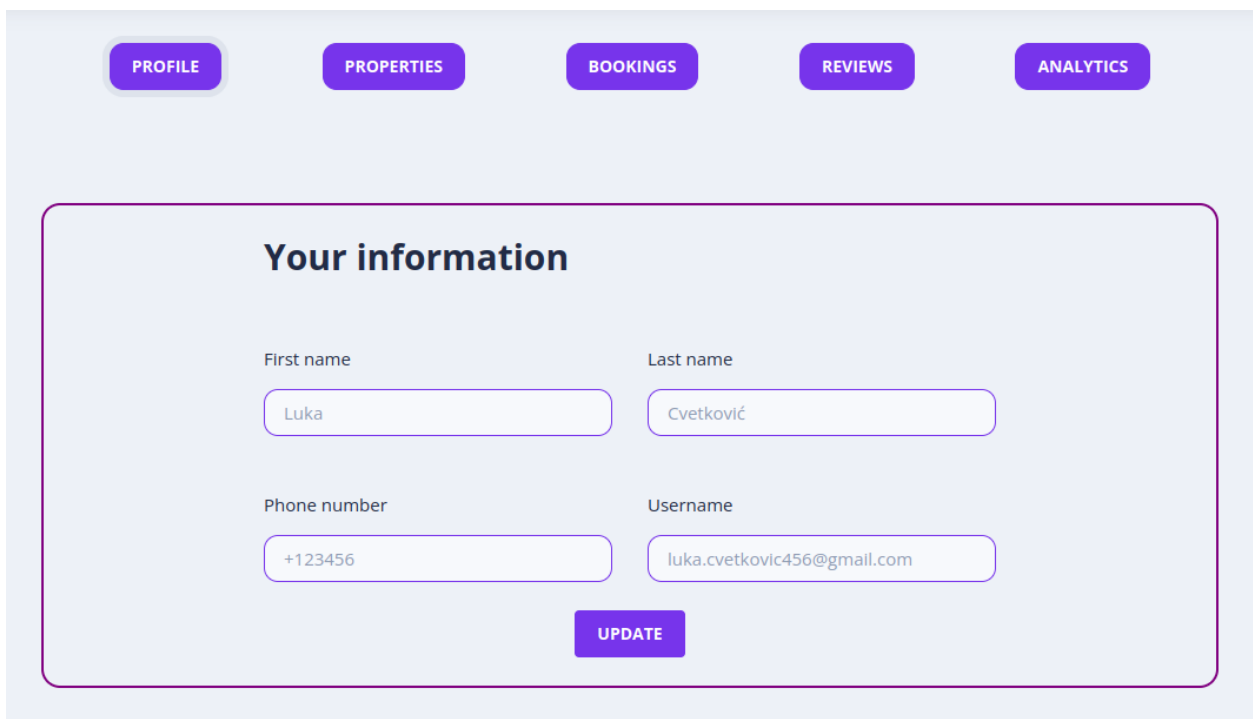
Password:

[CREATE ACCOUNT](#)

Already have an account? [Sign in](#)

**Slika 4.2.** Stranica za registraciju korisnika

Ako se korisnik prijavi kao administrator, otvara se komponenta `admin-panel`. Ona se sastoji od pet kartica od kojih mu svaka od njih daje pristup jednom dijelu podataka, a oni su: mogućnost dohvaćanja i uređivanja svojih informacija, mogućnost upravljanja svim smještajnim objektima, rezervacijama i recenzijama u aplikaciji, te uvid u analitičke podatke o smještajnim objektima. Slika 4.3.prikazuje izgled administratorove stranice u kojemu je on u mogućnosti urediti svoje podatke, te se također vide kartice, između kojih se administrator može navigirati. Kako bi se korisniku omogućila navigacija između kartica, koristi se funkcija `toggleTab` koju pokazuje Programski kod 4.7.(prikazan je samo dio funkcije zbog uštede prostora u radu). Ona se implementira na način da se uz pomoć `switch-case` izraza provjerava koja je kartica označena za otvaranje, te se jedino njena vrijednost postavlja na `true`, a sve ostale na `false`.



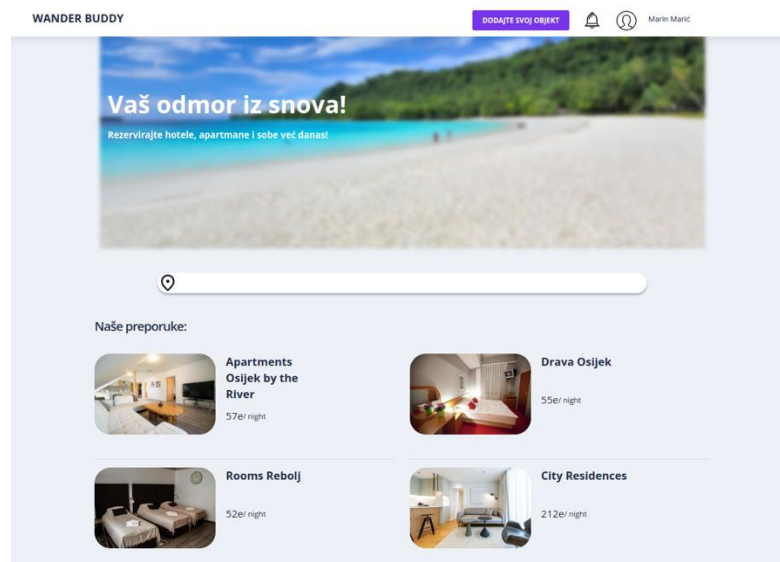
**Slika 4.3.** Stranica koja prikazuje izgled administratorske komponente

Ako se korisnik u aplikaciju prijavi kao standardni korisnik, otvara se komponenta pod nazivom `landing-page`, čiji izgled prikazuje Slika 4.4.Na vrhu ekrana se nalazi navigacijska traka koja se korisniku omogućuje kreiranje novog smještajnog objekta, prikaz svojih obavijesti, te klikom na svoje ime ili sliku, otvara profil korisnika. Ona se uvijek nalazi na vrhu ekrana, neovisno o tome u kojoj se korisnik trenutno komponenti nalazi. Naslovna stranica aplikacije se sastoji od preporuka smještajnih objekata, koji su fiksno odabrani iz API-ja, te polja za pretraživanje

objekata. Ovo polje omogućuje dohvaćanje svih smještajnih objekata iz baze podataka, ukoliko nije određen nijedan kriterij za pretraživanje.

```
async toggleTab(tabToOpen) {  
  switch (tabToOpen) {  
  
    case "profile":  
      this.viewProfile = true;  
      this.viewBookings = false;  
      this.viewFavorites = false;  
      this.viewProperties = false;  
      this.viewReviews = false;  
      break;  
  
    case "properties":  
      this.viewProfile = false;  
      this.viewBookings = false;  
      this.viewFavorites = false;  
      this.viewProperties = true;  
      this.viewReviews = false;  
      break;  
  
  }  
}
```

Programski kod 4.7. Funkcija za mijenjanje kartica na administratorovom profilu



Slika 4.4. Naslovna stranica standardnog korisnika

Programski kod 4.8. prikazuje logiku polja za pretraživanje. Funkcija `searchForParam` uzima tekst koji se upisuje u polje, te ukoliko je tekst duži od tri znaka, dohvaća sve objekte koji u sebi sadrže upisani dio ili cijelu lokaciju. Nakon dohvaćanja objekata iz poslužitelja, otvara se padajući niz sa prijedlozima te sa mogućnosti za pritisak tipke na opciju za izlistavanje više podudaranja. Klikom na tu opciju, otvara se komponenta `searched-results`, koju tumači Slika

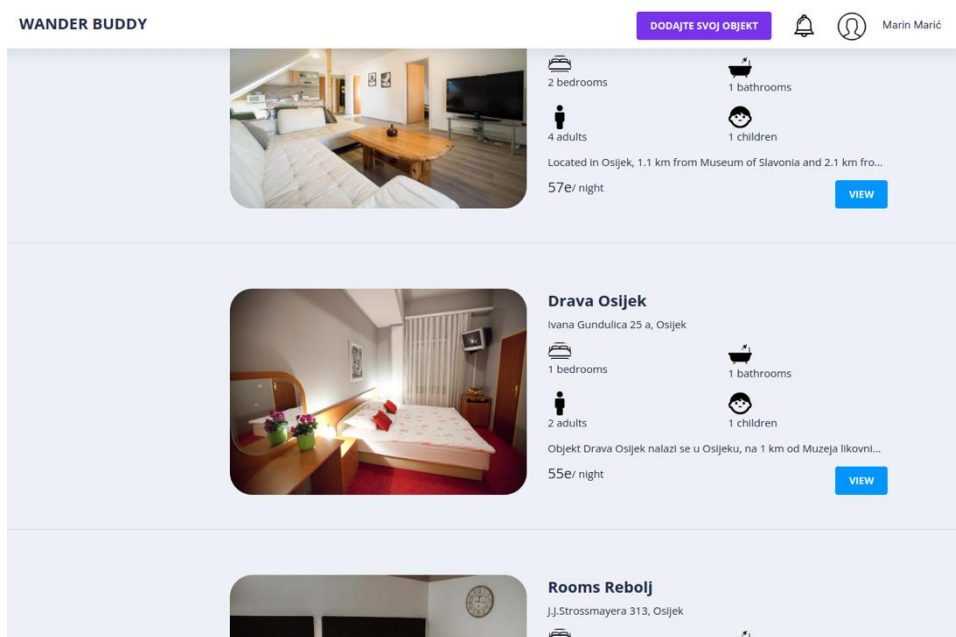


4.5. Na njoj se nalaze svi rezultati pretrage sa detaljima o pojedinom smještaju, te korisnik ima mogućnost otvoriti detalje o pojedinom smještajnom objektu klikom na tipku `view` ili na sliku smještajnog objekta.

```
searchForParam(param) {
  let searchParam = param.target.value;

  if(searchParam.length >= 3) {
    this._http.fullTextSearch(searchParam).subscribe((item) => {
      if(item) {
        console.log(item)
        this.displayedResults=item.slice(0,3);
        this.showDropdown=true;
        this.searchParam = searchParam;
      }
    })
  }
}
```

Programski kod 4.8. Funkcija za pretraživanje smještaja



Slika 4.5. Stranica koja prikazuje rezultate pretrage smještaja

Nakon otvaranja detalja o smještajnom objektu, pokreće se komponenta koja je ustvari podkomponenta naslovne stranice, a ona se naziva `view-location` i prikazuje ju Slika 4.6. Ona korisniku daje mogućnost detaljnog uvida u smještajni objekt, dodavanje objekta u listu svojih omiljenih objekata, te rezervaciju tog objekta za određeni datum. Također na pritisak tipke za

recenzije, korisnik ima mogućnost pogledati sve recenzije koje su prijašnji posjetitelji objekta ostavili, te ukoliko je prijavljeni korisnik odsjeo u tome smještaju, ima i sam mogućnost ostaviti recenziju te komentar. Kako korisnik nije u mogućnosti odmah vidjeti sve slike objekta, to će napraviti klikom na bilo koju od slika, te će se tada otvoriti skočni prozor koji daje mogućnost listanja svih slika odabranog objekta. Rezervacija smještaja se odvija tako da korisnik stisne gumb na dnu stranice nakon što izabere datume prijave i odjave, te broj gostiju. Datum prijave i odjave korisnika su u bazu podataka spremljeni kao Unix (engl. *Uniplexed Information and Computing System*) vremenski žig (engl. *timestamp*) [24]. On označava prošlo vrijeme u sekundama od određenog trenutka, poznatog kao vrijeme epohe koji obično počinje 1. siječnja 1970. godine, u ponoć po koordiniranom univerzalnom vremenu. Kako bi se vrijeme pretvorilo iz Unix vremenskog žiga u datum, koristi se funkcija `getNewDate` koju prikazuje Programski kod 4.9.

**Apartments Osijek by the River**  
Osijek, Croatia

**Brought to you, apartment by: marin\_maric**  
Max guests: 4  
Beds: 5  
Baths: 1

**Why us?**  
Located in Osijek, 1.1 km from Museum of Slavonia and 2.1 km from Museum Of Fine Arts in Osijek, Apartments Osijek by the River provides accommodation with free WiFi, air conditioning, free bikes and a garden. Featuring a kitchen with a microwave and a fridge, each unit also comes with a satellite flat-screen TV, ironing facilities, wardrobe and a seating area with a sofa. There is a fully equipped private bathroom with shower and a hairdryer. The apartment offers a terrace. Gradski Vrt Stadium is 2.4 km from the accommodation, while Osijek Citadel is 800 m away. The nearest airport is Osijek Airport, 20 km from Apartments Osijek by the River.

€ 57 [2 reviews](#)

Select Date:  
Sep 15, 2023 - Sep 22, 2023

1

1

€57 x 7 €399

Your total for 7 nights would be: €399

RESERVE NOW

Slika 4.6. Stranica koja pokazuje detalje smještaja

```

getNewDate(event) {
  console.log(event);
  let dateStart = event.start;
  let dateEnd = event.end;
  this.startMs = new Date(dateStart).getTime() / 1000;
  this.endMs = new Date(dateEnd).getTime() / 1000;
  let timeDiff = this.endMs - this.startMs;

  // To calculate the no. of days between two dates
  let dayDiff = timeDiff / (3600 * 24);
  this.dayDiff = dayDiff;
}

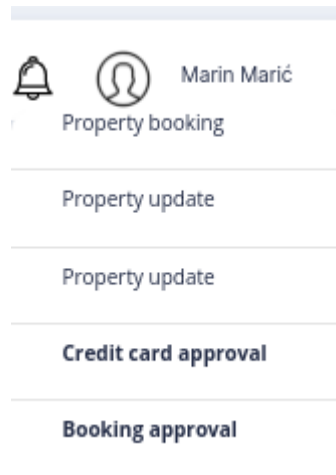
```

**Programski kod 4.9.** Funkcija koja pretvara Unix Timestamp u datum

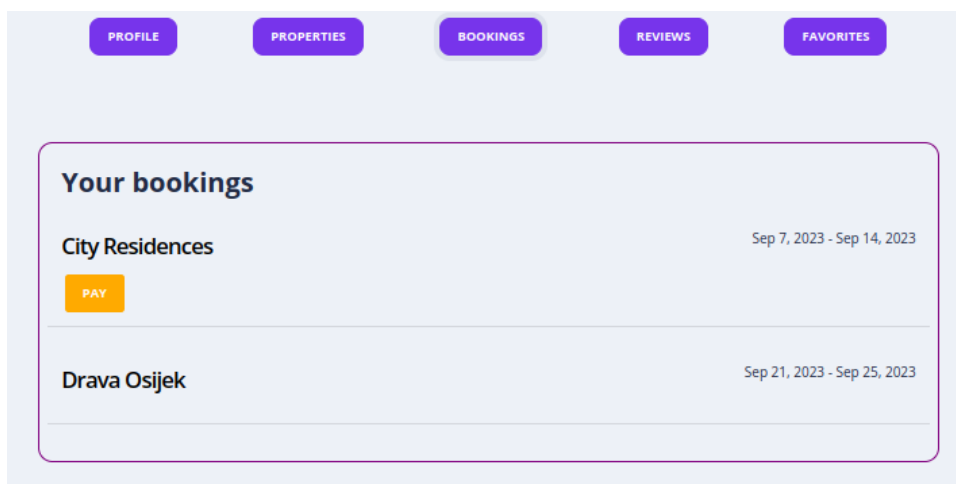
Kako je ranije napomenuto, korisnik u navigacijskoj traci ima mogućnost pogledati svoje obavijesti. Klikom na ikonu zvončića u traci, otvara se padajući niz koji prikazuje Slika 4.7., koji ispisuje sve obavijesti koje je korisnik primio od drugih korisnika ili administratora. Prilikom ispisa, provjerava se stanje obavijesti, te ako je polje `seen` označeno sa `false`, obavijest će biti ispisana podebljanim slovima, no ukoliko je poruka viđena, obavijest će se prikazati bez podebljanja. Korisnik može vidjeti obavijest klikom na nju, te će se njeno stanje automatski promijeniti u viđeno i više neće biti ispisano podebljanim slovima. Pored ikone zvončića nalaze se slika i ime korisnika. Klikom na njih otvara se izbornik sa pet kartica, isto kao u administratorovom prikazu, samo što se kod korisnika izbornik sastoji od:

- profila u kojemu može uređivati vlastite informacije te dodavati i uklanjati kreditne kartice koje služe za plaćanje objekata,
- izlistanja svih svojih smještajnih objekata koje može uređivati i brisati,
- izlistanja svih svojih rezervacija,
- prikaza svih recenzija koje je ostavio na tuđim objektima i
- smještajnih objekata koje je označio kao omiljene.

Ispis svih korisnikovih rezervacija prikazuje Slika 4.8. Ukoliko korisnik nije platio smještaj u objektu, ispod njega prikazuje se gumb koji mu omogućava plaćanje, te će se nakon što on podmiri iznos, ukloniti gumb te će jedna rezervacija izgledati kao gornji primjer na slici. Korisnik isto tako ima mogućnost i otkazati rezervaciju ukoliko nije u mogućnosti odsjesti u odabranom objektu u određenom vremenu.



Slika 4.7. Padajući niz korisnikovih obavijesti



Slika 4.8. Stranica koja prikazuje korisnikov profil

## 5. ZAKLJUČAK

U završnom radu razvijena je web aplikacija za upravljanje smještajnim kapacitetima čija je svrha bila unaprijediti rezervacije i organizacije smještajnih objekata. Rad je postavljen sa određenim ciljevima, što uključuje točno i detaljno proučavanje problema upravljanja smještajnim kapacitetima, razvijanje kvalitetnog korisničkog sučelja i implementaciju korisnih funkcionalnosti.

Prvo je opisan teorijski dio koji se odnosi na upravljanje smještajem i razvojem web aplikacije. Proučeni su ključni pojmovi smještajnih kapaciteta, rezervacije i organizacije smještaja. Zatim je razvijena funkcionalna web aplikacija koja se temelji na resurs-orijentiranom dizajnu. Aplikacija je podijeljena na korisnički i administratorski dio, te je tako prijavljenim korisnicima omogućeno rezerviranje smještaja i upravljanje svojim rezervacijama, dok administratori imaju pristup upravljanju smještajnim objektima, rezervacijama i samim korisnicima. U ovome su radu također detaljno opisane tehnologije korištene za razvoj aplikacije, uključujući Fastify i Angular te TypeORM za upravljanje bazom podataka. Ovo zajedničko korištenje tehnologija omogućilo je brzu i funkcionalnu izgradnju aplikacije.

Glavni ciljevi opisani u ovom radu su postignuti, uključujući implementaciju korisničkog sučelja, mogućnost rezervacije smještaja i organiziranje podataka o smještaju. Međutim, treba napomenuti neka ograničenja ovog rada. Iako je razvijena funkcionalna aplikacija, budući rad bi se mogao usredotočiti na optimizaciju performansi, posebno za velike skupove podataka. Također, posebnu pozornost treba obratiti na sigurnost aplikacije, uključujući zaštitu korisničkih podataka i zaštitu od mogućih napada. Ukratko, ovaj rad predstavlja dobar početak razvoja web aplikacije za upravljanje smještajem, postignuti su značajni rezultati, no moguća su daljnja poboljšanja i istraživanja. Ova aplikacija može olakšati proces rezervacije i organizacije smještaja, nudeći korisnicima jednostavan i učinkovit alat prilagođen njihovim potrebama.

## LITERATURA

- [1] Amazon Web Services. (n.d.). What Is an API? [online]. Amazon Web Services. Dostupno na: <https://aws.amazon.com/what-is/api/> [22.6.2023.].
- [2] Booking.com. (n.d.). [online]. Booking.com. Dostupno na: <https://www.booking.com/> [23.6.2023.].
- [3] Booking.com. Number One 3. [online]. Booking.com. Dostupno na: <https://www.booking.com/hotel/hr/number-one-3.hr.html> [23.6.2023.].
- [4] Airbnb. (n.d.). [online]. Airbnb. Dostupno na: <https://hr.airbnb.com/> [24.6.2023.].
- [5] Airbnb. Stay in Croatia - Listing ID 32079499. [online]. Airbnb. Dostupno na: <https://hr.airbnb.com/book/stays/32079499?numberOfAdults=2&checkin=2023-07-01&checkout=2023-07-09&numberOfGuests=2&guestCurrency=EUR&productId=32079499&isWorkTrip=false&numberOfChildren=0&numberOfInfants=0&numberOfPets=0> [24.6.2023.].
- [6] TripAdvisor. (n.d.). [online]. TripAdvisor. Dostupno na: <https://www.tripadvisor.com/> [24.6.2023.].
- [7] TripAdvisor. Tourism in Split, Croatia - Split Dalmatia County, Dalmatia. [online]. TripAdvisor. Dostupno na: [https://www.tripadvisor.com/Tourism-g295370-Split\\_Split\\_Dalmatia\\_County\\_Dalmatia-Vacations.html](https://www.tripadvisor.com/Tourism-g295370-Split_Split_Dalmatia_County_Dalmatia-Vacations.html) [24.6.2023.].
- [8] Vrbo - Vacation Rentals. (n.d.) [online]. Expedia Group. Seattle, Washington, SAD. Dostupno na: <https://www.vrbo.com/> [19.9.2023.].
- [9] Expedia - Travel Deals. (n.d.) [online]. Expedia Group. Seattle, Washington, SAD. Dostupno na: <https://www.expedia.com/> [19.9.2023.].
- [10] Wikipedia. (n.d.). Visual Studio Code [online]. Wikipedia. Dostupno na: [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code) [25.6.2023.].
- [11] Git. (n.d.). [online]. Git. Dostupno na: <https://git-scm.com/> [25.6.2023.].
- [12] S. Chacon, B. Straub, Pro Git, Creative Commons, California, USA, 2023.
- [13] Red Hat. (n.d.). What Is a REST API? [online]. Red Hat. Dostupno na: <https://www.redhat.com/en/topics/api/what-is-a-rest-api> [25.6.2023.].

- [14] L. Richardson, M. Amundsen, RESTful Web APIs, O'Reilly Media, Inc., California, USA, 2013.
- [15] Fastify. (n.d.). [online]. Fastify. Dostupno na: <https://www.fastify.io/> [25.6.2023.].
- [16] Google Cloud. (n.d.). API Design Guide: Resources [online]. Google Cloud. Dostupno na: <https://cloud.google.com/apis/design/resources> [25.6.2023.].
- [17] TypeORM. TypeORM - Amazing ORM for TypeScript and JavaScript [online]. TypeORM Community, 2021. Dostupno na: <https://typeorm.io/> [25.6.2023.]
- [18] MySQL AB. MySQL : MySQL Documentation [online]. Oracle Corporation, Santa Clara, CA, USA, 2021. Dostupno na: <https://dev.mysql.com/doc/> [25.6.2023.]
- [19] A. Beaulieu, Learning SQL, O'Reilly Media, Inc., California, USA, 2009.
- [20] Angular. (n.d.). [online]. Angular. Dostupno na: <https://angular.io/> [25.6.2023.].
- [21] Nebular. (n.d.). What is Nebular? [online]. Nebular. Dostupno na: <https://akveo.github.io/nebular/docs/getting-started/what-is-nebular#what-is-nebular> [25.6.2023.].
- [22] NordVPN. (n.d.). What Is bcrypt? [online]. Tefincom S.A. Dostupno na: <https://nordvpn.com/blog/what-is-bcrypt/> [4.9.2023.]
- [23] JWT.io. (n.d.). Introduction to JSON Web Tokens (JWT) [online]. Auth0, Inc. Dostupno na: <https://jwt.io/introduction> [4.9.2023.]
- [24] Unix Timestamp Converter [online]. Dostupno na: <https://www.unixtimestamp.com> [6.9.2023.]

## SAŽETAK

Završni rad objašnjava pojam i svrhu smještajnih kapaciteta. Govori o tome kako se upravlja smještajnim objektima, te njihovim značajkama i karakteristikama. Objašnjava proces rezervacije i organizacije smještaja sa korisničke strane. Detaljno je obrađeno postavljanje okruženja Visual Studio Code kako bi se omogućilo izrađivanje web aplikacije za upravljanje smještajnim kapacitetima. Navedeni su i pojašnjeni programski okviri Fastify, koji je korišten kako bi se izradila poslužiteljska strana aplikacije, te Angular, korišten za izradu klijentske strane aplikacije. Za oba su programska okvira navedene strukture obrazaca koje su korištene u pisanju koda. Konačno, prikazana je implementacija logike aplikacije u API dijelu projekta koji je pisan u TypeScript-u u programskom okviru Fastify. Nakon toga, prikazano je korištenje kreiranih ruta iz API dijela aplikacije u klijentskom dijelu koji je također pisan u TypeScript-u, te HTML-u i CSS-u uz pomoć programskog okvira Angular. Za kraj ovog završnog rada, objašnjeni su problemi predstavljeni prilikom definiranja aplikacije te na koji način ih izrađena aplikacija rješava.

Ključne riječi: Angular, Fastify, Rezervacija smještaja, Smještajni kapaciteti, Web aplikacija



## **ABSTRACT**

Development of a web application for accommodation management

The thesis explains the concept and purpose of accommodation capacity. It discusses how accommodation facilities are managed, including their features and characteristics. It explains the process of reservation and organization of accommodation from the user's perspective. Detailed instructions are provided for setting up the Visual Studio Code environment to enable the development of a web application for managing accommodation capacity. The Fastify framework is mentioned and explained, which was used to build the server-side of the application, as well as Angular, which was used for the client-side. The design patterns used in the code development for both frameworks are also described. Furthermore, the implementation of the application logic is demonstrated in the API part of the project, written in TypeScript using the Fastify framework. Subsequently, the utilization of the created API routes in the client-side, written in TypeScript, HTML, and CSS with the help of the Angular framework, is presented. Lastly, the thesis addresses the challenges encountered during the application's definition and explains how the developed application solves them.

Keywords: Angular, Fastify, Reservation process, Accommodation capacity, Web application

## **ŽIVOTOPIS**

Luka Cvetković, rođen je 18. Veljače 2002. u Našicama. Završio je svoje srednjoškolsko obrazovanje smjer komercijalist u Srednjoj školi Donji Miholjac. Godine 2020. Upisuje preddiplomski sveučilišni studij računarstva, smjer programsko inženjerstvo, na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku pri Sveučilištu Josipa Jurja Strossmayera u Osijeku. Tijekom svog obrazovanja zaposlio se u računalnoj tvrtki Beyondi d.o.o. u Osijeku gdje je i trenutno zaposlen. Tamo obavlja poslove backend developer-a u programskome okviru Fastify i poslove dizajna baze podataka, te dijagrama tokova koji određuju ponašanje aplikacije.

## **PRILOZI**

### **P.3.0. Izvorni kod aplikacije**

Dostupan na: <https://gitlab.com/luka.cvetkovic/wander-buddy>