

IZRADA WEB APLIKACIJE ZA OGLAŠAVANJE PRIVATNOG SMJEŠTAJA

Lončar, Ana

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:228:524410>

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

Download date / Datum preuzimanja: **2024-07-17**



Repository / Repozitorij:

[Repository of University Department of Professional Studies](#)



SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijska tehnologija

ANA LONČAR

ZAVRŠNI RAD

**IZRADA WEB APLIKACIJE ZA OGLAŠAVANJE
PRIVATNOG SMJEŠTAJA**

Split, lipanj 2023.

SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijska tehnologija

Predmet: Programiranje na internetu

ZAVRŠNI RAD

Kandidat: Ana Lončar

Naslov rada: Izrada web aplikacije za oglašavanje privatnog smještaja

Mentor: Marina Rodić, v. predavač

Split, lipanj 2023.

Sadržaj

1. Uvod	4
2. Korištene tehnologije	5
2.1 Django	5
2.2 Baza podataka SQLite	6
2.3 Bootstrap	6
3. Baza podataka i modeli	7
4. Korisničko sučelje	10
4.1 Prijava, odjava i registracija korisničkog računa	10
4.2 Naslovna stranica	12
4.3 Apartman	13
Korisničko sučelje apartmana	13
Vlasničko sučelje apartmana	17
Administratorsko sučelje apartmana	19
4.4 Moji apartmani	20
4.5 Dodavanje novog apartmana	20
4.6 Uređivanje apartmana	22
4.7 Brisanje apartmana	23
4.8 Nova cijena i uređivanje cijene	24
4.9 Brisanje cijene	25
4.10 Moji upiti	26
4.11 Uređivanje slika apartmana	28
4.12 Slanje upita apartmanu	31
4.13 Rezervacija apartmana	31
4.14 Odbijanje upita	33
4.15 Ostavljanje recenzije za apartman	33
4.16 Korisnički profil i njegovo uređivanje	34
4.17 Pretraga apartmana	35
5. Administratorsko sučelje	39

5.1	Početna stranica	39
5.2	Popis svih korisnika	39
5.3	Detalji o korisniku i njihovo uređivanje	41
5.4	Brisanje korisnika	42
5.5	Stvaranje novog apartmana	43
5.6	Ponovno aktiviranje izbrisanog apartmana	44
5.7	Ponovno aktiviranje izbrisanog korisnika	45
5.8	Administratorsko sučelje pregleda upita	45
5.9	Brisanje recenzija	46
5.10	Popis svih apartmana	46
6.	Zaključak	48
7.	Literatura	49

Sažetak

U ovom radu opisana je aplikacija za oglašavanje privatnog smještaja. Rađena je u radnom okviru (engl. *framework*) *Django*. Korisničko sučelje stilizirano je pomoću alata *Bootstrap*. U aplikaciji postoje dvije vrste korisničkih uloga, a to su korisnik i administrator. Cilj projekta je napraviti funkcionalnu aplikaciju koja se može koristiti kao smisljena cjelina te omogućuje sve osnovne usluge potrebne za iznajmljivanje i traženje apartmana.

Ključne riječi: apartman, django, mrežna aplikacija

Summary

Private accommodation application development

This work describes an application for advertising private accommodation. It was created using the Django framework. The user interface was designed with Bootstrap web toolkit. There are two kinds of roles a user can have in this application: user and administrator. The goal of this project is to make a functional application that makes sense as a whole and provides all the basic services needed for renting and finding apartments.

Keywords: apartment, django, web application

1. Uvod

Aplikacija koja je opisana u ovom radu nudi mjesto gdje iznajmljivači mogu reklamirati svoje apartmane, a gosti tražiti smještaj. Aplikacija ima dvije uloge: korisnik koji može iznajmljivati svoje apartmane i gostovati u tuđim apartmanima i administrator koji ne može biti vlasnik apartmana niti u njima boraviti, već je tu kao tehnička podrška.

U današnje vrijeme ovakve su aplikacije nužne. U prošlosti bi se ljudi uputili na ljetovanje bez rezervacija, računajući kako će uvijek biti slobodnog mjesta. No, danas gosti rezerviraju apartman mjesecima unaprijed, pa iznajmljivač više ne može dočekivati automobile ispred apartmana nadajući se kako traže smještaj. Mora se oglasiti svoj apartman. Za izradu ove aplikacije za motivaciju je služilo i to što autor osobno iznajmljuje smještaj preko web aplikacija. Većinu komunikacije s gostima autor obavlja preko email poruka, koje služe i kao obavijesti o aktivnosti, pa je i ova aplikacija tako dizajnirana. Zbog toga je važan kalendar zauzeća, jer to znatno smanjuje broj upita koji pokušavaju dobiti zauzeti termin.

U prvom poglavlju opisuje se sustav za upravljanje bazama podataka. To je jedan od prvih koraka pri planiranju projekta, jer ona definira kako se podaci mogu spremati i čitati, ograničavajući mogućnosti aplikacije.

U drugom poglavlju detaljno se opisuju opcije korisničkog sučelja. Korisničko sučelje je drugačije od administratorskog, s iznimkom stranice apartmana koja nema velike razlike, te je verzija njenog administratorskog sučelja opisana u ovom poglavlju.

U trećem poglavlju opisano je administratorsko sučelje. Administrator ima pravo vidjeti informacije o svim apartmanima i korisnicima te mijenjati njihove upite. On ima ovlast brisati korisnike i apartmane.

U zaključku se navodi rezultat ovog rada i nude primjeri na koji bi se način on mogao nadopuniti i usavršiti. Također se osvrće na programski okvir *Django* i dojam o njemu pri izradi ovog projekta.

2. Korištene tehnologije

2.1 Django

Aplikacija je rađena u mrežnom okviru *Django* koji je pisan u programskom jeziku *Python*. *Django* pruža programski okvir kako korisnik ne bi trebao ručno pisati kod koji se može automatizirati. Na taj način može se skratiti vrijeme izrade aplikacije i smanjiti broj greški radi manjeg ponavljanja koda. Ako na projektu odvojeno radi dizajner mrežnih (engl. *Web*) stranica, zbog smanjena količine koda na HTML stranicama (engl. *HyperText Markup Language*, tj. prezentacijski jezik za izradu mrežnih stranica) dizajner može raditi na njima bez straha kako će izmjenama napraviti problem. Programer također ne mora brinuti o zatvaranju veze s bazom podataka nakon svake transakcije, opet smanjujući mogućnost greški.

Django aplikacija podijeljena je po MVT strukturi na Model, Pogled i Predložak (engl. *Model, View, Template*). U datoteci `models.py` definiraju se tablice koje će postojati u bazi podataka. Svaka klasa predstavlja jednu tablicu, a unutar nje navode se stupci koje će ona imati i njihove karakteristike (npr. može li biti prazan, koji će se tip podataka spremati u njega, itd.). Na ovaj način izbjegava se direktni rad s SQL kodom (engl. *Structured Query Language*, tj. strukturni upitni jezik).

Datoteka `views.py` sadrži tzv. poslovnu logiku aplikacije, tj. odluke koje se moraju napraviti u kodu, npr. kome je dozvoljen pristup nekoj stranici ili koji će se podaci izvući iz baze podataka. Svaka funkcija je jedan Pogled i najčešće izvršava logiku za jednu stranicu. Pod pojmom Predložci smatra se skup HTML datoteka koje opisuju dizajn stranica. Pogled najčešće vraća jedan Predložak te mu šalje podatke koji će se prikazati na njemu. Za pomoć pri prikazivanju tih informacija koristi se poseban jezik za Predložke koji može raditi sa osnovnim logičkim naredbama.

Još jedan bitan element *Django* okruženja je `urls.py` datoteka. Ona zadaje koja će se funkcija Pogleda pozvati kad korisnik dođe na odgovarajuću URL poveznicu (engl. *Uniform Resource Locator*, tj. ujednačeni ili lokator sadržaja). Mogu se koristiti i varijable,

npr. identifikacijska oznaka (engl. *Id*) nekog podatka iz tablice čije će se informacije prikazati kada korisnik otvori stranicu s njegovom oznakom.

Pomoću ove strukture *Django* ima pristup odvojenim dijelovima aplikacije gdje svaki od njih obavlja svoju svrhu; Modeli definiraju bazu podataka, Pogled se bavi logikom preusmjerenja zahtjeva, a Predlošci predstavljaju korisničko sučelje. Na taj način se svaki od ovih dijelova može odvojeno mijenjati bez diranja ostatka aplikacije [1].

2.2 Baza podataka SQLite

Django okruženje podržava bazu podataka *SQLite*. To je sustav za bazu podataka koji je kompaktan, samostalan te ne traži posebnu konfiguraciju. Ne koristi odvojeni poslužiteljski proces, već direktno radi s datotekama na disku. Cijela baza podataka se nalazi u jednoj datoteci koja ne ovisi o platformi te se može otvoriti u bilo kakvom sustavu [2]. U ovoj aplikaciji korišten je *SQLite* 3.

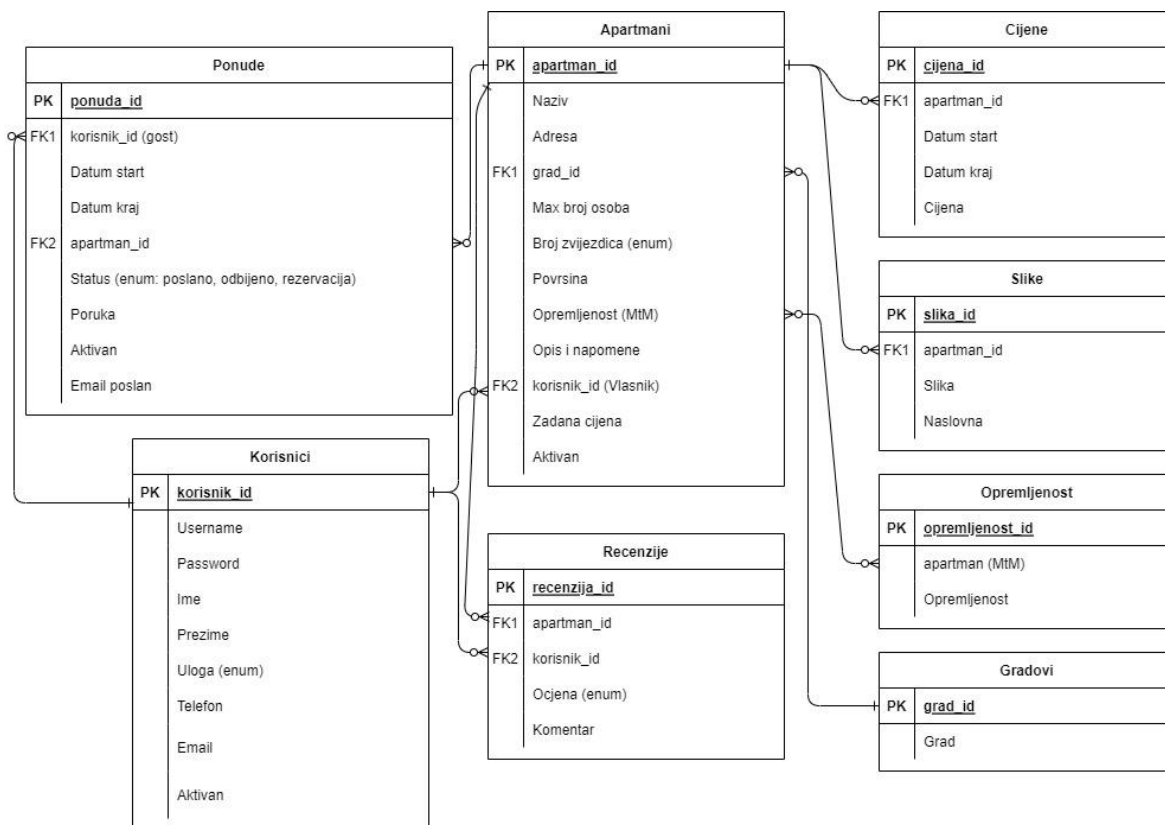
2.3 Bootstrap

Bootstrap je alat za lakše i moćnije uređivanje klijentskog dijela aplikacije (engl. *frontend*). Upotrebljava HTML, CSS (engl. *Cascading Style Sheets*, tj. stilski jezik) i *JavaScript*. Mogu se koristiti ugrađene komponente i koordinatne mreže te *JavaScript* dodaci bez korištenja *JQuery*. Koristi tehnologiju *Sass* za modularnu arhitekturu koja se može prilagođavati. Na taj način se može pisati vlastiti CSS uvozeći već postojeće komponente i varijable koje nam trebaju. Varijable se, na primjer, mogu koristiti ako se namjerava imati tematske boje za aplikaciju. Mogu im se dati imena koja će se pozivati u kodu kako bi se osiguralo korištenje istih boja. *Bootstrap* također koristi ugrađene CSS klase koje se mogu proizvoljno pozvati i međusobno kombinirati za oblikovanje nekog elementa i za definiranje njegove pozicije, veličine, boja i efekata [3]. Na raspolaganju su i *Bootstrap teme* koje su drugi korisnici dizajnirali i kreirali. Ova aplikacija bazirana je na temi "*Real Estate HTML Template*" od dizajnera *HTML Codex* (htmlcodex.com), čija licenca dopušta korištenje i mijenjanje teme.

3. Baza podataka i modeli

Sustav za upravljanje bazama podataka služi kako bi se informacije koje korisnik ili administrator unese mogle spremati ili dohvatiti. Praktički sve aplikacije, osim vrlo jednostavnih, trebaju bazu podataka. Baza je posebno važna ako aplikacija nudi ulogu korisničkih računa kako bi korisnikov profil i podaci ostali spremljeni.

Baza podataka se sastoji od tablica koje sadrže stupce. Svaki redak u tablici je zasebna informacija koja se može čitati, spremati, mijenjati ili brisati. Tablice mogu imati međusobne veze na druge tablice, npr. slika ima vezu na apartman kojemu pripada. Kod dizajniranja baze podataka treba obratiti pažnju kako bi se se iz nje mogli iščitati svi podaci koje će biti potrebno prikazati u aplikaciji. S druge strane, dobro je ne ponavljati informacije u bazi podataka, tj. raditi nepotrebne tablice.



Slika 1 - Baza podataka

Baza podataka ove aplikacije sastoji se od slijedećih tablica: Korisnici, Opremljenost, Gradovi, Apartmani, Cijene, Ponude, Recenzije i

Slike (Slika 1). Tablica Korisnici ima polja za korisničko ime, ime, prezime, email, telefon, ulogu (koja može biti korisnik ili administrator) te polje koje označava je li korisnik aktivan. Posljednje polje je dodano kako se korisnici ne bi trebali fizički brisati iz baze podataka, već se umjesto brisanja označe kao neaktivni, te se tako isključe iz bilo kakve pretrage korisnika kad to bude potrebno. Na taj način statistika može ostati očuvana.

Tablice Opremljenosti i Gradovi samo sadrže stupac s nazivom opremljenosti ili grada. Na ovaj način korisnik neće morati ručno upisivati grad, već će ga moći izabrati iz padajućeg izbornika te će se spriječiti unos varijacije istog naziva grada. Isto vrijedi i za opremljenosti, tj. dodatne mogućnosti koje apartman može nuditi, kao bazen, parking, dozvola za kućne ljubimce i slično.

Tablica Apartmani sastoji se od stupaca za naziv apartmana, adresu, maksimalni kapacitet ljudi, broj zvjezdica, površinu, kratak opis, zadanu (engl. *default*) cijenu i oznaku je li aktivan, koja služi istoj svrsi kao i kod korisnika. Zadana cijena je tu zato što uvijek mora biti dostupna neka cijena ako korisnik ne unese zasebnu cijenu za svaki period. Tablica Apartmani također sadrži vezu na tablicu Korisnik u stupcu koji označava vlasnika apartmana te na tablicu Grad kako bi se mogla unijeti lokacija apartmana. Apartman može imati samo jednog vlasnika, ali jedan korisnik može imati više apartmana (veza jedan na više). Isto vrijedi i za grad. Uz to, tablica Apartmani ima i vezu na tablicu Opremljenosti kojih može imati više. Jedna vrsta opremljenosti može biti uključena u mnogo apartmana (veza više na više).

Tablica Cijene sadrži stupce koje označavaju početni datum perioda, krajnji datum perioda i iznos cijene u Eurima. Ima i poveznicu na tablicu Apartman, kojeg može imati samo jednog, a apartman može imati više cijena.

U tablici Ponude postoje polja za početak i kraj perioda boravka, kratku poruku za vlasnika, status ponude (poslano, odbijeno ili rezervirano) te oznake je li ona aktivna i je li email obavijest za ostaviti recenziju poslana. Također tablica ima vezu na gosta koji je zatražio ponudu preko tablice Korisnici i apartman za koji je ponuda poslana preko tablice Apartmani (obje veze su jedan na više). Uzevši u obzir kako

ponude koje gost pošalje i rezervacije nemaju razlike u podacima koji su potrebni, i ponude i rezervacije se spremaju u istu tablicu, s tim što se označava radi li se o rezervaciji ili o ponudi (i je li ta ponuda odbijena). Oznaka za email služi kako bi se se moglo znati je li email koji korisnika podsjeća da ostavi recenziju poslan kako se ne bi slao više puta. U daljnjem tekstu za ponude će se koristiti riječ upiti radi veće jasnoće.

Tablica `Recenzije` sadrži polja za ocjenu koju korisnik daje apartmanu (od 1 do 10), komentar te ima vezu na tablicu `Apartman` za koji je ostavljen i tablicu `Korisnik` za onoga tko ju je ostavio (obje su jedan na više).

Konačno, tablica `Slike` sadrži sliku, vezu na apartman kojemu pripada (jedan na više) te oznaku je li ona naslovna slika tog apartmana kako bi se znalo koja će se slika prikazivati uz informacije o apartmanu.

4. Korisničko sučelje

4.1 Prijava, odjava i registracija korisničkog računa

Kako bi mogao pristupiti aplikaciji, korisnik mora otvoriti račun pomoću kojega će se spremati informacije vezane za njega. Svaka stranica, osim onih za registraciju i prijavu, je zaštićena dekoratorom `@login_required` koji korisnika preusmjerava na stranicu za prijavu ako pokuša pristupiti aplikaciji ako nije prijavljen.

Preko korisničkog sučelja mogu se stvarati samo računi sa ulogom `Korisnik`, dok se za administratorske račune očekuje kako neće biti napravljeni preko korisničkog sučelja, već direktno u bazi podataka.

Kod registracije, korisnik upisuje željeno korisničko ime (engl. `username`), lozinku, email, ime i prezime te broj telefona. Kada potvrdi unos podataka u Pogledu mu se automatski dodjeljuje uloga korisnika te ga se preusmjerava na stranicu za prijavu (ispis 1).

```
if request.method == 'GET':

    form = MyUserCreationForm()

    return render(request, 'register.html', {"form":form})

elif request.method == 'POST':

    form = MyUserCreationForm(request.POST)

    if form.is_valid():

        post = form.save(commit=False)

        post.uloga = 'korisnik'

        post.save()

        return redirect('http://localhost:8000/login/')
```

Ispis 1 – Registracija korisnika

Kod prijave, korisnik upisuje svoje korisničko ime i lozinku te ako su podaci ispravni, preusmjeri ga se na početnu stranicu.

Kada se korisnik odjavi, prije nego što se odjava odradi vrši se provjera je li potrebno poslati email podsjetnike za recenziju. Oni se šalju nakon što korisnikov boravak u apartmanu završi kako bi ga se podsjetilo na ostavljanje recenzije. To znači da se, prilikom odjave, traže rezervacije gdje je korisnik gost i čiji je datum završetka prošao. Uzimaju se u obzir samo oni upiti koji su aktivni i koji nemaju oznaku kako je email već poslan. Potom se za svaki od njih šalje email gostu sa podsjetnikom za ostavljanje recenzije sa poveznicom pomoću koje je može ostaviti. Na kraju, tim upitima se postavlja oznaka kako je za njih email poslan (ispis 2).

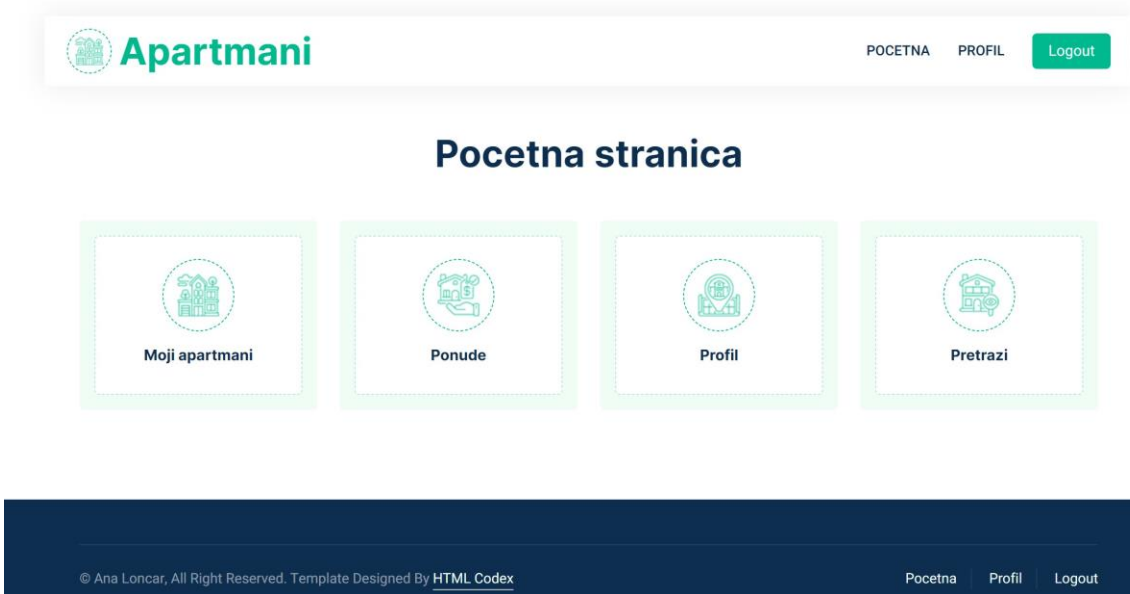
```
mail_due = Ponude.objects.filter(aktivan=True).filter(status="rez")
            .filter(gost=korisnik).filter(datum_end__lte=today).filter(mail_sent=False)
            for rez in mail_due:
                ...

                send_mail(
                    subject='Ne zaboravite ostaviti recenziju za apartman ' +
                    apartman.naziv,
                    ...)

                rez.mail_sent = True
                rez.save()
```

Ispis 2 – Provjera za podsjetnik pri odjavi korisnika

4.2 Naslovna stranica



Slika 2 - Naslovna stranica

Naslovna stranica (slika 2) sadrži glavni izbornik gdje korisnik može vidjeti koje su mu opcije dostupne te njima pristupiti. Izbornik običnog korisnika i administratora se razlikuje. Kada korisnik pristupi ovoj stranici, njegova uloga se provjerava te mu se prikazuje prikladni izbornik. Ako je korisnik izbrisan (tj. označen kao neaktivan), nakon što se prijavi i bude preusmjeren na ovu stranicu, njega se odmah odjavljuje, zabranjujući mu pristup aplikaciji (ispis 3).

```
if korisnik.aktivan == False:
    return redirect('http://localhost:8000/logout/')
if korisnik.uloga == "korisnik":
    return render(request, 'pocetna.html', {"korisnik":korisnik})
elif korisnik.uloga == "admin":
    return render(request, 'admin_pocetna.html', {"korisnik":korisnik})
```

Ispis 3 – Preusmjeravanje kod naslovne stranice

Opcije u izborniku dostupne korisniku su njegovi apartmani, upiti, profil i pretraga.

4.3 Apartman

Korisničko sučelje apartmana

Karakteristike stranice apartman ovise o više stvari: koju ulogu ima korisnik (admin ili korisnik), je li korisnik vlasnik apartmana te je li apartman deaktiviran.

Ako je korisnik vlasnik apartmana ili je administrator, prebacuje ga se na verziju stranice koja omogućava administratorske privilegije (više o tome poslije). Ako apartman nije aktivan, korisniku su sve poveznice pomoću kojih bi mogao doći do svoje stranice nedostupne. Također je bilo potrebno osigurati kako se direktnim upisivanjem URL poveznice ne bi moglo pristupiti nedozvoljenom sadržaju. Ako korisnik to ipak pokuša napraviti, umjesto stranice apartmana pojavi mu se poruka "Taj apartman više ne postoji" (ispis 4).

```
if apartman.aktivan == False:
    return HttpResponse("Taj apartman više ne postoji!")
if korisnik == apartman.vlasnik or korisnik.uloga == "admin":
    return redirect('http://localhost:8000/apartmanvlasnik/' +
str(apartman_id))
```

Ispis 4 – Provjera pri ulasku na sučelje apartmana

Na vrhu stranice (slika 3) prikazan je naziv apartmana, njegova slika te osnovne informacije iz baze podataka - adresa i grad, površina, maksimalni broj gostiju koji može primiti, kategorija u kojoj pripada (npr. apartman sa 3 zvjezdice) te opis i napomene koje je vlasnik dodao. Ako apartman nema niti jednu sliku, na mjesto naslovne slike apartmana stavlja se grafika koja predstavlja praznu sliku, tj. činjenicu kako slika ne postoji.

Ispod slike i informacija nalazi se opcija pomoću koje korisnik šalje upit tom apartmanu (slika 3).



Sv. Lucija

- 📍 Ulica Slobode 14, Omis
- 🏠 78.00 m²
- 👤 4 Osoba
- ★ Kategorija: 5 Zvij.

Opis i napomene:

Posalji upit

Slika 3 - Opis apartmana

Navedene su sve vrste opremljenosti koje apartman može imati (npr. okućnica, bazen itd.) te pokraj svake od njih nalazi se kvačica ukoliko je apartman opremljen s tom pogodnosti ili znak X ako nije (slika 4).

- | | |
|---|--|
| <ul style="list-style-type: none"> ✗ Parking ✗ Bazen ✓ Okucnica ✗ Rostilj ✗ Klimatizacija ✗ Terasa | <ul style="list-style-type: none"> ✗ Internet ✗ Dozvoljeni kućni ljubimci ✓ TV ✗ Stednjak ✗ Perilica rublja ✗ Perilica posuda ✗ Mikrovalna |
|---|--|

Slika 4 - Popis opremljenosti

Kako bi se to postiglo, uspoređuju se tri liste: sve opremljenosti, opremljenosti koje apartman ima i opremljenosti koje nema. Zadnja lista je na početku postavljena kao prazna, pa se određivanjem elemenata koji nisu u listi opremljenosti apartmana puni s tim elementima (ispis 5).

```

ap_opr = apartman.opremljenosti.all()
sve_opr = Opremljenost.objects.all()
bez_opr = []
for so in sve_opr:
    if so not in ap_opr:
        bez_opr.append(Opremljenost.objects.get(id = so.id))

```

Ispis 5 – Popunjavanje tri liste opremljenosti

Na Predlošku provjerava se nalazi li se svaka od opremljenosti u jednoj ili drugoj listi i u skladu s time stavlja se prikladna oznaka (ispis 6).

```

{% for so in sve_opr1 %}

    {% for ao in ap_opr %}

        {% if so.id == ao.id %}

            <i class="fa fa-check text-primary me-2"></i>

        {% endif %}

    {% endfor %}

    {% for bo in bez_opr %}

        {% if so.id == bo.id %}

            <i class="fa fa-times text-secondary me-2"></i>

        {% endif %}

    {% endfor %}

    {{so.opremljenost}}

{% endfor %}

```

Ispis 6 – Određivanje oznake opremljenosti

Na stranici su također vidljive i osnovne informacije o vlasniku apartmana, kako bi ga korisnik mogao kontaktirati. Na stranici za uređivanje profila korisnika se upozorava kako će ti podaci biti vidljivi na njegovim apartmanima. Nakon toga nalazi se galerija slika (slika 5). Korisnik može kliknuti na pojedinu sliku kako bi se ona otvorila u novom prozoru u punoj veličini.

Kontakt vlasnika:



Slika 5 - Kontakt vlasnika i galerija

Korisniku je dostupan kalendar na kojemu su crvenom bojom označeni zauzeti termini (dani) za apartman (slika 6). Odabirom "Prošli mjesec" i "Slijedeći mjesec" korisnik može mijenjati prikaz mjeseca. Kao URL parametri šalju se mjesec, godina i sidro koje služi kako bi se korisnika pozicioniralo na kalendar kada on mijenja mjesec. Ako se prime svi parametri, koriste se vrijednosti koje su se dobile u daljnjem kodu. U suprotnom, korisnik je prvi puta došao na stranicu (tj. nije još kliknuo na prošli ili slijedeći mjesec), te se koristi trenutna godina i mjesec (ispis 7).

```
if request.GET.get('godina') and request.GET.get('godina'):  
    godina = int(request.GET.get('godina'))  
    mjesec = int(request.GET.get('mjesec'))  
else:  
    godina = date.today().year  
    mjesec = date.today().month
```

Ispis 7 – Određivanje mjeseca i godine

Ovi parametri moraju biti primljeni HTTP metodom `Get`. Ako je korisnik pritisnuo dugme "Prošli mjesec", u HTTP metodi `Post` postojati će parametar "Prošli". U većini slučajeva tada se mjesec uvećava za 1, ali ako je mjesec Siječanj, godina se treba smanjiti za jedan, te mjesec postaviti na 12. Analogno tome se izvršava kod za slijedeći mjesec. S ovim se dobiva mjesec (i godina) koji se treba prikazati (ispis 8).

```
if 'Prosli' in request.POST:
    if mjesec == 1:
        godina = godina-1
        mjesec = 12
    else:
        mjesec = mjesec-1
```

Ispis 8 – Rubni slučaj godine i mjeseca

Ti parametri, zajedno sa listom svih rezervacija tog apartmana, se šalju funkciji koja je zaslužna za formatiranje kalendara korištenjem `HTMLCalendar` paketa koji je ugrađen u HTML 5. Funkcija prolazi kroz svaki dan te provjerava je li taj dan zauzet, današnji dan ili prazan i boja ga odgovarajućom bojom.

Pokraj toga je prikazana informacija o cijenama apartmana (Slika 6). Prvo je napisana zadana cijena, koja se koristi u slučaju ako nije navedena cijena za taj period. Potom je prikazan kompletan cjenik smještaja, kategoriziran prema vremenskim razdobljima.

U zadnjem dijelu stranice nalaze se recenzije i ocjena za apartman. Iz svih ocjena koje taj apartman ima računa se prosjek. Ako nema ocjena prosjek se neće prikazati. Za pojedinačne recenzije prikazuje se ocjena tog korisnika, njegovo ime i prezime te komentar koji je ostavio.

Vlasničko sučelje apartmana

Ako je korisnik vlasnik tog apartmana, ali nema ulogu administratora, uz prethodno spomenute, ima na raspolaganju i dodatne značajke. Na vrhu stranice nalazi se opcija za brisanje apartmana. Opcija za slanje upita nije dostupna. Podaci o vlasniku također nisu potrebni, pa se ne prikazuju. Vlasnik može urediti ili dodati novu sliku. Kod popisa cijena, pokraj svake cijene je opcija za njeno uređivanje ili brisanje. Ispod tablice postoji opcija za dodavanje nove cijene (slika 6).

Prošli mjesec

Sljedeći mjesec

Cijene:

Ako nije naveden period: €70.00

March 2023						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Period	Cijena	Opcije	
June 1, 2023 - June 30, 2023	€60.00	Uredi	Izbriši
July 1, 2023 - July 31, 2023	€75.00	Uredi	Izbriši
Aug. 1, 2023 - Aug. 31, 2023	€80.00	Uredi	Izbriši
Sept. 1, 2023 - Sept. 15, 2023	€50.00	Uredi	Izbriši

Nova cijena

↑

Slika 6 - Kalendar i cijene

Slijedeće, u dva stupca su podijeljeni upiti (isključivo za taj apartman). Kod dohvaćanja upita, bilo je bitno uzeti u obzir samo aktivne upite kako se upiti od izbrisanih korisnika ne bi računali. Rezervacije su odvojene u posebnom stupcu kako bi vlasnik mogao lako vidjeti koji su termini zauzeti u tom apartmanu (kronološki poredano), dok su svi ostali upiti (koji nisu još odgovoreni ili koji su odbijeni) u drugom stupcu (slika 7). Ako upit ima status "poslano", tj. na upit vlasnik nije odgovorio, može ga odbiti ili rezervirati (prihvatiti). Vlasnik nema mogućnost ponovno prihvatiti odbijene upite.

Rezervacije:

Ante Rozovic
 📅 March 4, 2023 - March 7, 2023
Otkazi

Marta Lukic
 📅 March 8, 2023 - March 9, 2023
Otkazi

Ponude:

Marta Lukic
 📅 Sept. 29, 2023 - Oct. 1, 2023

Status: ODBIJENO

Slika 7 - Rezervacije i ponude

Administratorsko sučelje apartmana

Ako je korisnik administrator, ima iste opcije kao i vlasnik apartmana, osim iznimki koje će biti navedene. Ako korisnik nije administrator, a preko URL poveznice pokušava pristupiti izbrisanom apartmanu, biti će mu javljeno kako taj apartman više ne postoji (ispis 9). Također će biti zabranjen pristup ovom sučelju ako nije ni vlasnik ni administrator (ako je jedno od to dvoje, može ući).

```
if korisnik != apartman.vlasnik and korisnik.uloga != "admin":  
    return HttpResponse("Nesto je poslo po krivu!")  
  
if apartman.aktivan == False and korisnik.uloga != "admin":  
    return HttpResponse("Taj apartman vise ne postoji!")
```

Ispis 9 – Provjere pri ulasku na stranicu apartmana

Administrator ima mogućnost vidjeti apartman koji je izbrisan, za razliku od običnog korisnika. Za podsjetnik, apartmani se nikada ne brišu iz baze podataka radi očuvanja statistike, već se samo deaktiviraju, čime se računaju kao izbrisani. Zastavica koja označava je li apartman izbrisan se šalje Predlošku. Kako bi odmah bilo jasno kako se radi o izbrisanom apartmanu, stranica takvog apartmana ima tamnu pozadinu i bijeli tekst. Umjesto opcije za brisanje apartmana biti će dostupna opcija za ponovno postavljanje apartmana kao aktivnog. Izbrisani apartman se ne smije mijenjati kako bi se spriječila manipulacija statistike, pa su sve opcije za uređivanje (slike, cijene, upiti, recenzije) onemogućene.

Za razliku od vlasnikovog sučelja, administrator može vidjeti podatke o vlasniku apartmana. Također klikom na ime vlasnika može doći do administratorske stranice za tog korisnika. Administrator može rezervirati odbijeni upit, dok vlasnik to ne može. Ako je apartman aktivan, administrator može izbrisati recenziju sa tog apartmana.

4.4 Moji apartmani

Korisnik na ovoj stranici može vidjeti popis svih svojih apartmana i pristupiti njihovim stranicama (slika 8). Ako je apartman izbrisan, on nije vidljiv korisniku na ovom popisu. Ako on ima ulogu administratora prebaciti će ga se na popis svih apartmana u aplikaciji u administratorskom sučelju.



Apartman	Opcije
Sv. Jure	Uredi Izbriši
Gospina Stina	Uredi Izbriši
Apartman Jelena	Uredi Izbriši

Slika 8 - Moji apartmani

Na vrhu stranice nalazi se opcija dodavanja novog apartmana (koji je u vlasništvu prijavljenog korisnika). U tablici s apartmanima korisniku je dostupno pristupiti svakom od apartmana klikom na njegovo ime. Također su za svaki apartman dostupne opcije uređivanja i brisanja.

4.5 Dodavanje novog apartmana

Korisnik ima mogućnost izrade svoga apartmana. Administrator nema pravo na vlastite apartmane i, ako pokuša aktivirati opciju upisivanjem URL poveznice u preglednik, zahtjev će mu biti odbijen.

Kao i kod svih formi, u kodu se razlikuje slučaj kada korisnik tek otvori stranicu s namjerom upisivanja podataka (tj. dođe do nje HTTP metodom `Get`) i kada nakon što je upisao podatke njih potvrdi pritiskom na “Potvrdi” (HTTP metoda `Post`). U prvom slučaju, učitava se forma za apartman i Predlošku se prosljeđuje lista svih opremljenosti kako bi ih se moglo ponuditi kao opcije za označiti (ispis 10).

```
if request.method == 'GET':  
  
    sve_opr = Opremljenost.objects.all()  
  
    form = ApartmanForm()  
  
    return render(request, 'apartman_forma.html', {"form":form,  
"sve_opr":sve_opr})
```

Ispis 10 – Opremljenosti apartmana u formi

Kada korisnik klikne na “Potvrdi”, podaci koje je korisnik upisao se dobivaju pomoću forme, a posebno se dobiva lista opremljenosti koje je korisnik označio pomoću kvačica. Ako su podaci ispravni, forma se još uvijek ne sprema u bazu, dok se ne postave još neke opcije; uzima se prijavljeni korisnik i njega se postavlja kao vlasnika apartmana. Korisnik, naime, nema opciju na formi za biranje vlasnika jer to može biti samo on i očekuje se kako će se u kodu to i postaviti. Nakon toga odmah se iz baze dobavlja novostvoreni apartman i prolaskom kroz listu opremljenosti koje je korisnik označio dodaje ih se tom apartmanu i ponovno ga se sprema u bazu podataka (ispis 11). Potom se korisnika preusmjerava na popis njegovih apartmana.


```

elif request.method == 'POST':

    form = ApartmanForm(request.POST)

    nove_opr = request.POST.getlist('opremljenost')

    if form.is_valid():

        post = form.save(commit=False)

        post.vlasnik = korisnik

        post.save()

        apartman = Apartmani.objects.get(id = post.id)

        for opr_id in nove_opr:

            opr = Opremljenost.objects.get(id = opr_id)

            apartman.opremljenosti.add(opr)

        apartman.save()

```

Ispis 11 – Forma za apartmane u Pogledu

4.6 Uređivanje apartmana

Korisnik može urediti vlastiti apartman, ali administrator ima to pravo nad svim apartmanima (osim ako je apartman izbrisan). To znači kako samo u slučaju ako korisnik nije ni vlasnik ni administrator on nema pravo urediti taj apartman. Također, ako apartman nije aktivan, dobit će poruku kako taj apartman više ne postoji.

Za uređivanje apartmana koristi se slična forma kao i za njegovo kreiranje. Kad korisnik otvori formu, u njoj već moraju biti upisani podaci tog apartmana, a to uključuje da opremljenosti koje ima moraju biti označene. Zbog toga se Predložku šalju podaci apartmana u formu te popis opremljenosti koje apartman već ima. U Predložku se prolaskom kroz listu svih mogućih opremljenosti svaka od njih pojedinačno traži u listi opremljenosti tog apartmana te ju se, ukoliko je pronađena, postavlja kao označenu (ispis 12).

```

{% for so in sve_opr %}
    <input type="checkbox" name="opremljenost" value="{{so.id}}"
        {% for ao in ap_opr %}
            {% if so.id == ao.id %} checked {% endif %}
        {% endfor %}>
    {{so.opremljenost}} <br>
{% endfor %}

```

Ispis 12 – Označavanje već odabranih opremljenosti

Kada korisnik klikne na “Potvrdi” nije potrebno korisnika postaviti kao vlasnika, jer je on već postavljen pri kreaciji i ne može se mijenjati. Ali, zato što se opremljenosti mogu ne samo dodavati nego i skidati, apartman se čisti od prijašnjih opremljenosti koje je imao. Potom se svaka od novo zadanih opremljenosti njemu pridružuje. Korisnika se preusmjerava na stranicu tog apartmana (ispis 13).

```

if form.is_valid():
    apartman.opremljenosti.clear()

    for opr_id in nove_opr:
        opr = Opremljenost.objects.get(id = opr_id)
        apartman.opremljenosti.add(opr)

    apartman.save()
    form.save()

```

Ispis 13 – Pridruživanje novih opremljenosti apartmanu

4.7 Brisanje apartmana

Izbrisati apartman ima pravo njegov vlasnik ili administrator. Ako nije ni jedno od toga, neće moći pristupiti ovoj opciji preko URL poveznice ako je sam upiše, a na stranici apartmana koji nije njegov ionako neće imati opciju brisanja. Bitno je podsjetiti kako se brisanjem apartmana njemu samo daje oznaka neaktivnosti te kako se nikad zapravo ne briše iz baze podataka. Također je potrebno izbrisati (tj. označiti kao neaktivne) sve upite vezane za taj apartman, kako oni ne bi zbunjivali korisnike (ispis 14).

```

if 'Izbrisi' in request.POST:

    if Ponude.objects.filter(aptman = aptman).exists():

#sve ponude od apartmana

        pon_izbrisi = Ponude.objects.filter(aptman = aptman)

        for pon in pon_izbrisi:

            pon.aktivan = False

            pon.save()

    aptman.aktivan = False

    aptman.save()

```

Ispis 14 – Brisanje apartmana

Kada korisnik želi izbrisati apartman, prvo ga se dovodi na stranicu gdje mora potvrditi kako je siguran želi li obaviti brisanje. Ako kao odgovor aplikacija dobije "Izbriši", apartman će se označiti kao neaktivan, kao i njegove ponude. Korisnika se onda preusmjerava na listu njegovih apartmana. Samo administrator može ponovno aktivirati izbrisani apartman.

4.8 Nova cijena i uređivanje cijene

Uz postavljanje opće cijene za pojedini apartman, korisnik može odrediti i različite cijene za određene vremenske periode. Ako za traženi period nije definirana cijena uzima se u obzir opća cijena, koja je obavezna za unijeti pri kreaciji apartmana (i može se mijenjati poslije). Nema granice koliko se cjenovnih perioda može zadati te se periodi ne ponavljaju (tj. vrijede samo za godinu koju korisnik unese). Cijenu može unijeti samo vlasnik tog apartmana ili administrator i to ako je apartman aktivan. Nakon što korisnik potvrdi unos na formi, cijeni se pridružuje apartman u bazi podataka.

Potrebno je provjeriti preklapa li se period cijene s nekom drugom cijenom tog apartmana. Treba osigurati sve moguće situacije: početak nove cijene je prije početka stare a kraj nove je usred perioda stare, početak nove cijene je usred perioda stare a kraj nove je

nakon kraja stare, početak nove cijene je prije početka stare a kraj nove je nakon kraja stare i cijeli period nove cijene je unutar perioda stare. Za sve ove situacije dovoljne su tri provjere: nalazi li se početak nove cijene usred perioda stare, nalazi li se kraj nove cijene usred perioda stare, te nalazi li se cijela nova cijena usred perioda stare cijene. Ako je bilo koje od toga potvrdno, korisnikov zahtjev se odbija i obavještava ga se kako taj period već ima cijenu (ispis 15).

```
new_start_date = post.datum_start
new_end_date = post.datum_end

clashing = Cijene.objects.filter(aptman=aptman).filter(
    Q(datum_start__lte=new_start_date, datum_end__gte=new_start_date) |
    Q(datum_start__lt=new_end_date, datum_end__gte=new_end_date) |
    Q(datum_start__gte=new_start_date, datum_end__lte=new_end_date))

if clashing.exists():
    return HttpResponse("Taj period vec ima cijenu!")
```

Ispis 15 – Provjera zauzetosti perioda

Također, provjerava se nije li korisnik stavio početni datum prije završnog, te je li završni datum u prošlosti, što se ne dozvoljava. Korisnika se na kraju preusmjerava na stranicu apartmana kojemu cijena pripada.

Korisnik može urediti, tj. promijeniti cijenu. Također je i ovdje bitno obaviti sve provjere kako se cijene ne bi preklapale. No, treba prvo isključiti iz traženja samu tu cijenu kako se ona ne bi uspoređivala sama sa sobom. Korisnika se potom vraća na apartman.

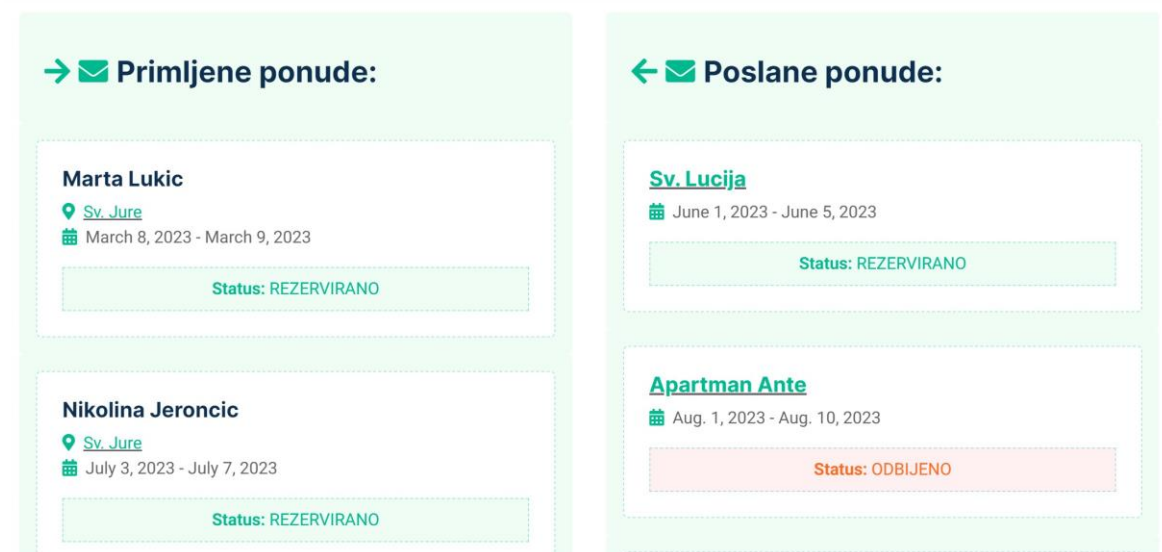
4.9 Brisanje cijene

Cijene se, za razliku od npr. apartmana, skroz brišu iz baze podataka. Slično kao i za apartman, prijavljeni korisnik mora biti vlasnik za apartman na kojega se odnosi cijena ili mora biti administrator te vezani apartman mora biti aktivan. Korisniku se nudi opcija odustajanja od brisanja. Ako ipak odabere "Izbriši", još jedan put se provjerava ima li

pripadajuće ovlasti (zbog opasnosti od upisivanja URL poveznice) te se cijena briše iz baze podataka.

4.10 Moji upiti

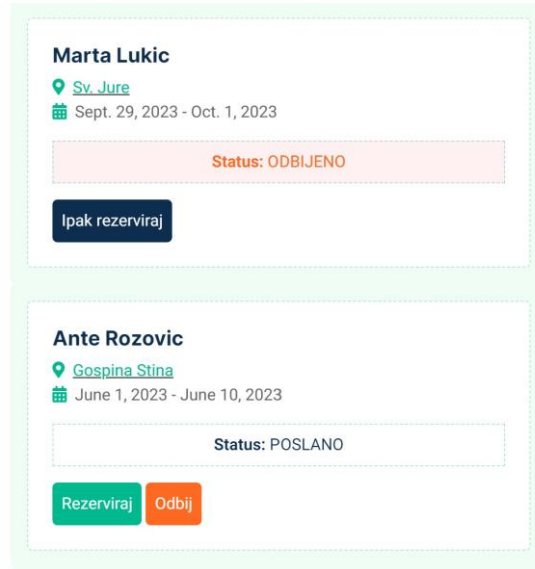
Na ovoj stranici korisnik može lako vidjeti sve upite koji ga se tiču (slika 9). U lijevom stupcu nalaze se upiti koji su poslani njegovim apartmanima od strane drugih korisnika, dok se u desnom stupcu nalaze upiti koje je korisnik poslao drugim apartmanima. U oba stupca prikazuje se 10 zadnjih upita, s novijim upitima na vrhu. Na dnu svakog stupca je opcija za vidjeti sve ponude koje pripadaju tom stupcu. Bitno je što se neaktivni upiti ne prikazuju; upit postaje neaktivan kad se vezani apartman izbriše te kad administrator izbriše korisnika koji je poslao upit ili čijem je apartmanu poslan upit. Upiti označeni kao neaktivni smatraju se izbrisanima, iako se nikad zapravo ne brišu iz baze podataka.



Slika 9 - Upiti

U upitima koji je korisnik primio ispisane su informacije kao ime i prezime potencijalnog gosta, naziv apartmana kojemu je poslan upit (klikom na njega otvara se stranica apartmana) te početni i krajnji datum kada je gost zatražio biti u apartmanu. Razlikuju se tri tipa upita: rezervirani, odbijeni i poslani. Pod poslanim upitom se smatra upit na koji vlasnik apartmana nije još odgovorio. U tom slučaju vlasnik može odabrati

opcije "Odbij" ili "Rezerviraj". Ako je upit odbio, može se još uvijek predomisli i kliknuti na "Ipak rezerviraj" (slika 10). Rezervacije su konačne i samo ih administrator može poništiti.



Slika 10 - Priljeni upiti

Kod upita koje je sami korisnik poslao drugim apartmanima, naveden je naziv apartmana (klikom na kojeg se prebacuje na stranicu apartmana), vremenski period koji korisnik želi rezervirati i status upita, koji su isti kao i kod lijevog stupca. Ako je korisnik odležao u apartmanu, ima pravo njemu napisati recenziju. To pravo se provjerava gledajući je li upit rezerviran te je li krajnji datum rezervacije u prošlosti (ispis 16).

```
{% if pon.status == "rez" %}
    {% if pon.datum_end < datum %}
        <a href="..."> Napiši recenziju </a>
    {% endif %}
{% endif %}
```

Ispis 16 – Prikaz opcije pisanja recenzije

U pogledu svih upita (poslanih ili primljenih) upiti su podijeljeni po stranicama kako bi se ograničio broj upita koji je prikazan odjednom. Iz Predložka se prenosi trenutni broj stranice na koju se želi doći te se u Pogledu dohvaćaju podaci na toj stranici (ispis 17).

```
paginator = Paginator(primljene_ponude,5)
page_number = request.GET.get('page')
page_obj = paginator.get_page(page_number)
return render(request, 'primljene_ponude.html', {"primljene":page_obj})
```

Ispis 17 – Podjela na stranice

4.11 Uređivanje slika apartmana

Ako je apartman aktivan ispod galerije slika prikazivati će se opcija za uređivanje slika. Također će biti dostupna i opcija za dodavanje nove slike, iako se njoj može pristupiti i na samoj stranici za uređivanje slika.

Urediti slike može vlasnik apartmana ili administrator te je opcija zaštićena od bilo kojeg drugog neautoriziranog posjeta. Iako opcija neće biti prikazana ako je apartman izbrisan, to se isto provjerava pri pristupu stranice kako ga netko ne bi pokušao izmijeniti preko URL poveznice.

Mogućnosti uređivanja slika apartmana su dodavanje nove slike, brisanje te mijenjanje naslovne slike (slika 11). Naslovna slika je ona slika koja se prikazuje na vrhu stranice apartmana uz njegove informacije te pri kratkom opisu apartmana u opciji pretrage. Samo jedna slika u apartmanu može biti naslovnica. Pored svake slike nalazi se kvačica (engl. *radio button*) za odabir nje kao naslovnju sliku. Kako bi se, kad korisnik tek uđe u uređivanje slika, kvačica uredno prikazivala kao označena pokraj trenutne naslovnice, Predložku se šalje podatak o staroj naslovnoj slici iz baze podataka. U Predložku se prolaskom kroz sve slike svaka slika uspoređuje sa starom naslovnicom te, ako je ista, postavlja se kao označena (ispis 18)

```

<form method="GET">
    {% for slika in slike %}
        
        <input type="radio" name="nova_naslovna" value="{{slika.id}}"
        {% if slika.id == naslovnaid %} checked {% endif %}>
            Naslovna Slika
        <a href="http://localhost:8000/confdelslika/{{slika.id}}">
            Izbrisi </a>
    {% endfor %}
    <input type="submit" value="Potvrdi naslovnu sliku">
</form>

```

Ispis 18 – Uređivanje slika u Predlošku

Ako se promijeni naslovnica, kako bi se taj podatak spremio potrebno je odabrati "Potvrdi naslovnu sliku". Tada Pogled zaprimi informacije o novoj naslovnoj slici. Ako je odabrana naslovnica drugačija slika nego bivša naslovnica, ta slika će postati označena kao naslovna slika. Također se staroj naslovnici briše ta oznaka (ispis 19).

```

if nova_naslovna != stara_naslovna:
    ostale_slike=slike.exclude(id=nova_naslovna_id)
    nova_naslovna.naslovna = True
    nova_naslovna.save()
    for slika in ostale_slike:
        slika.naslovna = False
        slika.save()

```

Ispis 19 – Provjera za naslovnicu

Pokraj svake slike također je dostupna opcija za njeno brisanje (ispis 18). Ako se klikne, korisnik, ako ima određena prava, mora potvrditi kako je to stvarno mislio učiniti. Ako odgovori potvrdno, ponovno se provjerava se ima li prava to napraviti, tj. je li administrator ili vlasnik apartmana i je li apartman aktivan. Zatim, prije nego se slika izbriše, važno je napraviti provjere u vezi naslovne stranice. Na primjer, ako je slika koju briše bila naslovnica, iduća slika po redu se postavlja kao naslovnica kako bi apartman uvijek imao naslovnu sliku. Ako je brisana slika bila jedina, to jest apartman neće više imati sliku, u tom slučaju se ne može postaviti nova naslovnica te će se umjesto nje prikazivati grafika koja označava praznu sliku (ispis 20).

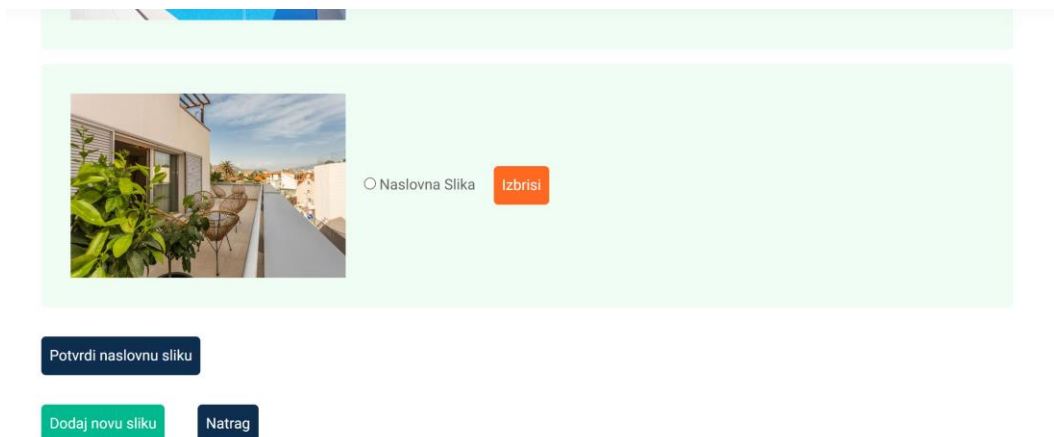

```

if 'Izbrisi' in request.POST:
    if slika.naslovna == True:
        slike = Slike.objects.filter(aptman__id = aptman_id)
            .exclude(id=slika_id)
        nova_naslovna = slike.first()
        if nova_naslovna is not None: #ako nije zadnja slika
            nova_naslovna.naslovna = True
            nova_naslovna.save()
        slika.delete()

```

Ispis 20 – Brisanje slike

Zadnja prikazana opcija je za dodavanje nove slike (slika 11). Nakon provjera ovlasti korisnik može odabrati sliku sa svog računala i postaviti je. Kao i kod brisanja, treba se provjeriti hoće li se novo dodana slika tretirati kao naslovnica. U slučaju ako je to prva slika, to jest apartman prije nje nije imao ni jednu sliku, ona postaje naslovnica, kako bi naslovna slika postojala čak i ako korisnik nikad sam ne bira tu opciju. U suprotnom, u bazi podataka se zapisuje kako ona nije naslovnica. Slika se potom sprema u bazu podataka.



Slika 11 - Uredi slike

4.12 Slanje upita apartmanu

Na stranici apartmana, čiji on nije vlasnik, korisnik može poslati upit za razdoblje kada se nada rezervirati apartman. Ako pokuša rezervirati svoj apartman direktnim pisanjem URL poveznice, to će mu biti odbijeno. Ako administrator proba napraviti istu stvar, isto to neće moći. Također se obavlja provjera je li apartman izbrisan.

U formi za upit korisnik mora unijeti početni i krajnji datum svog boravka i ako želi može napisati kratku poruku vlasniku, koja će mu biti poslana putem elektroničke pošte.

U Pogledu se automatski za upit veže korisnik koji je prijavljen, apartman na kojem je korisnik kliknuo "Pošalji upit" te se upit postavlja kao aktivan. Treba se provjeriti je li taj vremenski period već rezerviran (ako je samo upit poslan za taj period, a nije još rezerviran, on se ne računa). Također se, naravno, računaju samo aktivni upiti. Provjera se obavlja na isti način kao i kod unosa cjenovnih perioda. Slično tome, također se provjerava je li početni datum prije krajnjeg, te jesu i početni i krajnji datum u budućnosti.

Kada se upit spremi u bazu podataka, vlasniku apartmana se šalje email obavijest kako je dobio novi upit. U njemu su navedeni naziv apartmana, ime i prezime gosta, adresa apartmana, vremenski period te poruka korisnika ako je ima. Također se, iz razloga što se ne može odgovoriti na email adresu s koje se šalju obavijesti, vlasnika se obavještava kako za daljnju komunikaciju gosta treba kontaktirati na njegovu email adresu, koja je također navedena.

4.13 Rezervacija apartmana

Apartman se smatra rezerviranim kada vlasnik prihvati upit koji je poslan njegovom apartmanu. Rezervaciju može poništiti samo administrator.

U trenutku prihvaćanja upita izrađuje se račun boravka u apartmanu koji se gostu šalje na email. Kod računanja ukupne cijene, uzima se svaki dan zasebno u periodu boravka gosta i provjerava postoji li posebno navedena cijena koja obuhvaća taj dan. Ako ne, uzima se zadana cijena apartmana. Svi dani se pribroje u ukupnu cijenu, koja se u

računu šalje zajedno s imenom gosta, nazivom apartmana, adresom apartmana, imenom vlasnika i njegovom email adresom, datumima boravka te ukupnim brojem dana koje će gost provesti u apartmanu (ispis 21).

```
delta = timedelta(days=1)

while start_date <= end_date:

    if Cijene.objects.filter(apartman_id = apartman.id)

        .filter(datum_start__lte=start_date,

datum_end__gte=start_date).exists():

            cijena = Cijene.objects.filter(apartman_id =

                apartman.id).get(datum_start__lte=start_date,

                datum_end__gte=start_date).cijena

        else:

            cijena = apartman.default_cijena

        ukupno += cijena

        uk_dana += 1

        start_date += delta

    ...

ponuda.status = "rez"

ponuda.save()
```

Ispis 21 – Računanje cijene boravka

Upit može prihvatiti samo vlasnik tog apartmana ili administrator. Uz to što apartman koji je zaprimio upit mora biti aktivan, također se provjerava je li i sam upit aktivan. Zadnja provjera koju je važno napraviti je uvjeriti se ponovno kako je taj period slobodan, jer, iako nije bio zauzet u vremenu slanja upita, u međuvremenu se mogao rezervirati nekom drugom gostu.

Jedina promjena koja se radi nad upitom u bazi podataka je što mu se kao status postavi rezervacija (ispis 21).

4.14 Odbijanje upita

Vlasnik ili administrator može odbiti upit kojeg je poslao gost ako mu iz bilo kojeg razloga ne odgovara. Ako se predomisli, može ga ipak odlučiti rezervirati.

Kako bi mogao odbiti upit, korisnik mora biti vlasnik apartmana ili administrator te apartman i upit moraju biti aktivni, isto kao i kod rezervacije.

U bazi podataka upit se označava kao odbijen te se gostu šalje email obavijest kako mu upit nije prihvaćen. U emailu su navedeni naziv i adresa apartmana te podaci o vlasniku s napomenom kako za daljnju komunikaciju treba kontaktirati vlasnika na njegovu email adresu.

4.15 Ostavljanje recenzije za apartman

Nakon što završi boraviti u apartmanu gostu je dozvoljeno ocijeniti taj apartman recenzijom. Ona uključuje ocjenu od 1 do 10 i komentar. Očekuje se kako će najčešći način pristupanja ovoj opciji biti preko poveznice u emailu koji korisnik dobije kao podsjetnik nakon što mu istekne period boravka. Opcija je također dostupna i u popisu korisnikovih poslanih upita ako je boravak završio.

Zato što joj se pristupa uglavnom preko poveznice, za ovu je opciju posebno važno osigurati kako bi samo osobe koje imaju pravo na nju mogle ostaviti recenziju. Administrator nema pravo na ostavljanje recenzija. Također se provjerava je li apartman aktivan i da korisnik nije vlasnik tog apartmana. Korisnik može samo jedan put ocijeniti isti apartman, pa se provjerava postoji li već korisnikova ocjena u bazi podataka. Zatim, mora se potvrditi kako postoji rezervacija (a ne samo upit) na korisnikovo ime za taj apartman. Ako postoji, korisnik je možda više puta bio gost u tom apartmanu, pa se uzima u obzir posljednji. Za njega se potom provjerava je li datum odlaska već prošao, tj. je li gost završio boravak u apartmanu (ispis 22).

```
pon_1 = Ponude.objects.filter(aptman__id = apartman_id).filter(gost__id =
korisnik_id).order_by('-id')[:1]

for p in pon_1:

if p.datum_end > date.today():

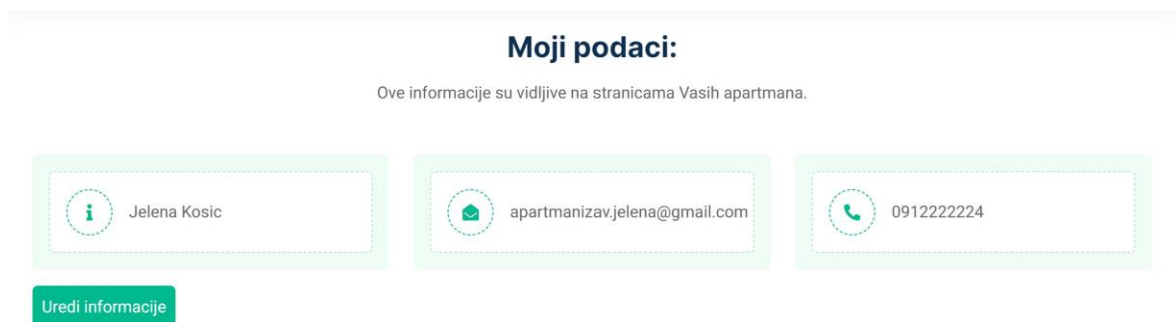
return HttpResponse("Niste jos boravili u tom apartmanu!")
```

Ispis 22 – Provjera boravka u apartmanu

U formi recenzije samo se ispunjava ocjena i komentar, a prijavljenog korisnika i apartman se u Pogledu pridružuje recenziji. Nakon što se ona spremi u bazu podataka, vlasniku na email dolazi obavijest kako ima novi upit. Navodi se i gostov email za daljnju komunikaciju, zajedno s njegovim imenom i nazivom apartmana za koji je zaprimljen upit.

4.16 Korisnički profil i njegovo uređivanje

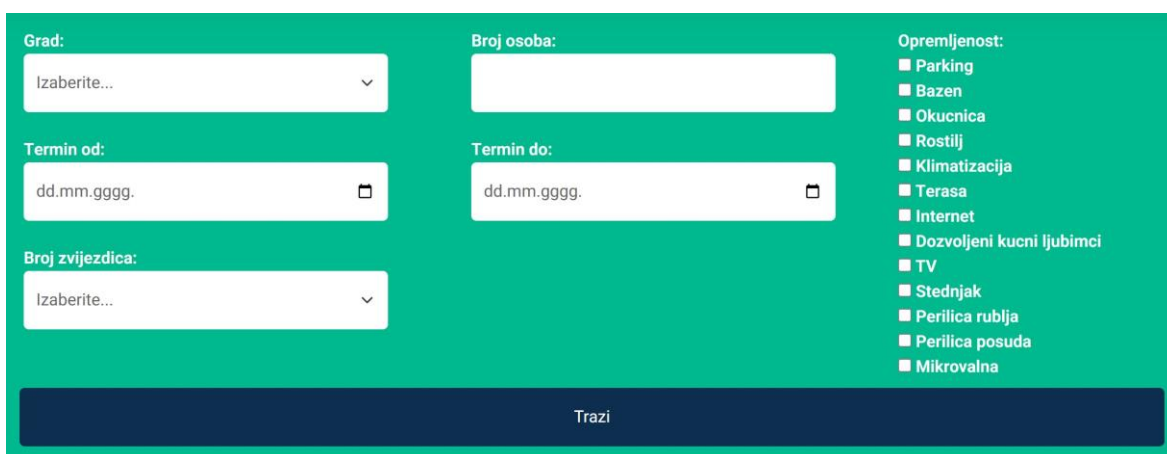
Profil je u ovoj aplikaciji jednostavan: sastoji se od imena i prezimena, email adrese te broja telefona (slika 12). Ove informacije su važne za kontakt s gostima ili vlasnicima apartmana. Ako je korisnik vlasnik apartmana, ove informacije se pojavljuju na stranicama njegovih apartmana. Korisnikov email će biti naveden u određenim email obavijestima kako bi se omogućila daljnja komunikacija, jer se ne može odgovarati na email s kojeg je obavijest poslana. Korisnik može vidjeti i urediti svoje podatke na svom profilu.



Slika 12 - Profil

4.17 Pretraga apartmana

Pretraživanje apartmana je glavni način na koji korisnici mogu otkriti i pristupiti stranicama apartmana koji nisu njegovi. Korisnici mogu suziti odabir određivanjem parametara u bilo kojoj kombinaciji (tj. dozvoljeno je upisivanje više njih odjednom), i to: biranjem grada, koliko osoba će boraviti u apartmanu, vremenskog perioda kada planira boraviti (kako bi se prikazali samo apartmani koji su tada prazni), kategorije apartmana (tj. broja zvjezdica) te označavanjem koje dodatne opremljenosti apartman mora imati (slika 13).



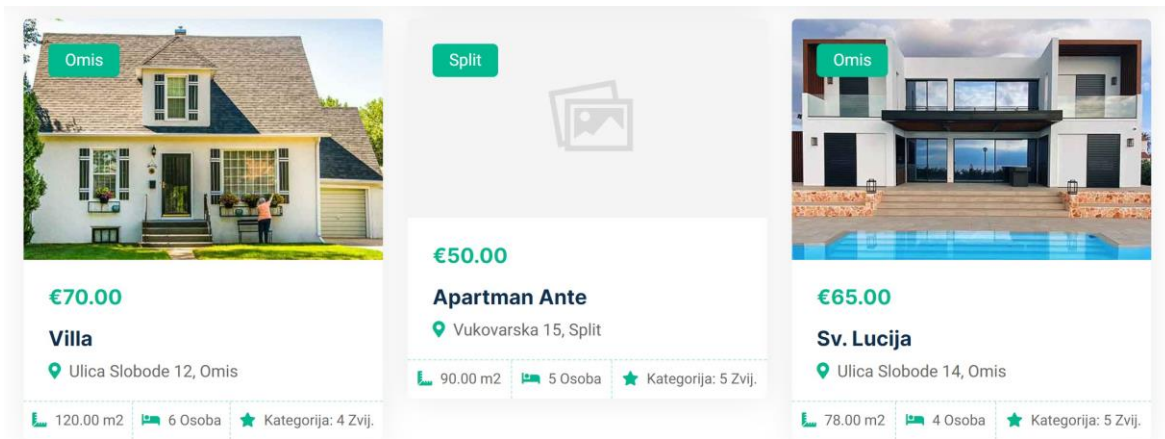
The image shows a search interface for apartments. It features a teal background with white input fields and a dark blue search button. The search criteria are organized into three columns:

- Grad:** A dropdown menu with the placeholder text "Izaberite...".
- Broj osoba:** A text input field.
- Termin od:** A date input field with the placeholder "dd.mm.gggg." and a calendar icon.
- Termin do:** A date input field with the placeholder "dd.mm.gggg." and a calendar icon.
- Broj zvijezdica:** A dropdown menu with the placeholder text "Izaberite...".
- Opremljenost:** A list of amenities, each with a small square checkbox:
 - Parking
 - Bazen
 - Okucnica
 - Rostilj
 - Klimatizacija
 - Terasa
 - Internet
 - Dozvoljeni kućni ljubimci
 - TV
 - Stednjak
 - Perilica rublja
 - Perilica posuda
 - Mikrovalna

At the bottom center, there is a dark blue button with the white text "Traži".

Slika 13 - Parametri pretraživanja

Apartmani se potom pojavu u obliku kartica sa kratkim informacijama o njemu (slika 14). Tu je naslovna slika (ako je ima, a u suprotnom je grafika) na kojoj je naglašen grad u kojem se nalazi i cijena boravka, njegov naziv, adresa te konačno površina, maksimalni kapacitet gostiju i kategorija. Ako mu je apartman zanimljiv, korisnik može klikom na njegovo ime ili naslovnicu doći na stranicu apartmana gdje može vidjeti detaljnije informacije.



Slika 14 - Rezultati pretraživanja

U Pogledu se isprva iz baze podataka uzimaju svi apartmani koji su aktivni, a potom se filtriraju u ovisnosti koje je parametre korisnik upisao. Svaki od parametara koji se provjeravaju u kodu može biti preskočen, a oni koji nisu će iz liste apartmana izbaciti apartmane koji više ne odgovaraju. Na taj način se u bilo kojoj kombinaciji lista apartmana može čistiti dok se ne dođe do rezultata koji je korisnik tražio. Ako korisnik nije odabrao ni jedan parametar, zastava će ostati na `False` i lista apartmana će se postaviti kao prazna, kako bi se izbjeglo vađenje svih apartmana iz baze podataka. Ako je, međutim, bilo koji od parametara upisan, zastava će se postaviti na `True` (ispis 23).

Za broj osoba uzimaju se samo oni apartmani kojima je maksimalni kapacitet veći od tog broja. Za broj zvjezdica i grad treba se također provjeriti kako opcija koja je odabrana nije "Izaberite ..." jer je to opcija koja dočeka korisnika kad otvori stranicu. Za slobodno vremensko razdoblje gledaju se samo rezervacije koje su aktivne i provjerava se preklapaju li se datumi na sličan način kao i kod slanja upita. Ako se preklapaju, tada taj apartman u to vrijeme nije slobodan i isključuje ga se iz liste. Za opremljenosti apartmana kod svakog apartmana prolazi se kroz popis opremljenosti koje je korisnik zadao. Za svaku od tih opremljenosti provjerava se je li je apartman ima. Ako bilo koji od njih nema, apartman se isključuje iz popisa (ispis 23).

```

flag = False
...
if is_valid_param(opremljenosti) and opremljenosti!=[]:
    for apartman in apartmani:
        for opr_id in opremljenosti:
            if apartman.opremljenosti.filter(id=opr_id).exists():
                continue
            else:
                apartman_id = apartman.id
                apartmani = apartmani.exclude(id=apartman_id)
        flag = True
...
if (flag==False):
    apartmani = ""

```

Ispis 23 – Primjer provjere parametara tražilice

Kako bi se ispravno pokazala slika, odvojeno se rade popisi apartmana bez naslovnice te lista svih slika koje imaju oznaku naslovnica (jedna po apartmanu). Nakon što prođe sve filtere, u listi apartmana se provjerava ima li pojedini apartman uopće ikakvih slika (jer jedino ako ih nema neće imati naslovnicu) i ako nema stavlja ga se u popis apartmana bez naslovnice. U Predlošku, kada se apartman ispisuje na ekran, provjerava se je li on u listi apartmana bez naslovnice. Ukoliko je, na njenom mjestu prikazuje se grafika koja označava praznu sliku. U suprotnom, u popisu svih naslovnica traži se ona koja je povezana s tim apartmanom i prikazuje se (ispis 24).


```

{% for ap in bez_naslovne %}

    {% if apartman.id == ap.id %}

        <a href="http://localhost:8000/apartmangost/{{apartman.id}}">

        </a>

    {% endif %}

{% endfor %}

{% for naslovna in naslovne %}

    {% if apartman.id == naslovna.apartman.id %}

        <a href="http://localhost:8000/apartmangost/{{apartman.id}}">

        </a>

    {% endif %}

{% endfor %}

```

Ispis 24 – Naslovnica apartmana u Predlošku

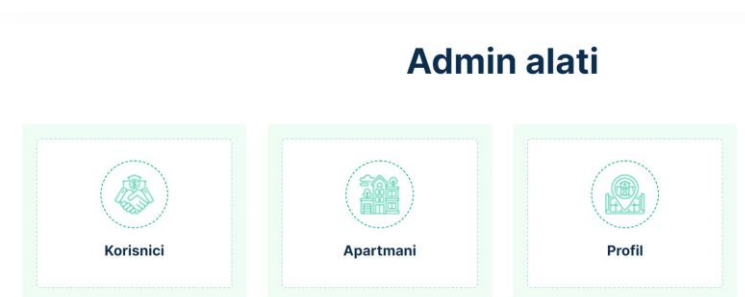
Na Predložak se također mora poslati i lista svih gradova i mogućih opremljenosti. Gradovi se prikazuju u padajućem izborniku, a opremljenosti kao kvačice koje korisnik može označiti.

Pretraga može imati više stranica ako nađenih apartmana bude više od šest. Kako bi parametri pretrage ostali očuvani kroz šetanje stranicama koristi se oznaka Predloška (engl. *template tag*) `query_transform`. To je korisnički napisan dodatak Predlošku koji uzima stare parametre koji su bili poslani stranici na kojoj se nalazimo i dodaje im nove.

5. Administratorsko sučelje

5.1 Početna stranica

Kad se korisnik prijavi u aplikaciju, preusmjerava ga se na početnu stranicu gdje mu se prema njegovoj ulozi daje odgovarajući izbornik. Ako korisnik ima ulogu administratora bit će mu ponuđen administratorski izbornik, koji se sastoji od popisa korisnika, apartmana te njegovog profila (slika 15). Pri ulasku na svaku stranicu administratorskog sučelja provjerava se ima li korisnik ulogu administratora.



Slika 15 - Naslovnica za administratore

5.2 Popis svih korisnika

Na ovoj stranici administrator može pregledati i pristupiti svim korisnicima (slika 16). Prikazuju se, naravno, samo oni kojima je uloga Korisnik, a ne Admin, jer administrator nema pravo gledati ni mijenjati druge administratore.

Ime: Prezime: Username: E-mail:

Traži

Ime i prezime	Username	Email	Opcije
Marta Lukic	marta123	apartmanizav.jelena@gmail.com	<input type="button" value="Uredi"/> <input type="button" value="Izbrisi"/>
Nikolina Jeroncic	nikolina123	apartmanizav.jelena@gmail.com	<input type="button" value="Uredi"/> <input type="button" value="Izbrisi"/>

Slika 16 - Popis korisnika

Administrator može tražiti određenog korisnika ili korisnike po imenu, prezimenu, korisničkom imenu (engl. *username*) i emailu. Na sličan način kao i kod pretrage apartmana, ovi se filteri mogu kombinirati na bilo koje načine jer se inicijalni popis svih korisnika sužava sa svakim parametrom pretrage koji je administrator unio. Također se korisnike može abecedno sortirati prema imenu i prezimenu, korisničkom imenu i emailu klikom na naziv odgovarajućeg stupca tablice. Ako nije ništa odabrano, korisnici se poredaju od novijih prema starijima. Nakon što se popis sortira, naziv stupca po kojem je sortiran se ponovno šalje u Pogled. Tako on neće biti podcrtan, naznačujući kako je trenutno odabran te kako nema potrebe na njega kliknuti (ispis 25).

```

<a href="?{% query_transform order='ime' %}"
{% if order_by != 'ime' %} text-decoration: underline; {% endif %}>
Ime i prezime</a>

```

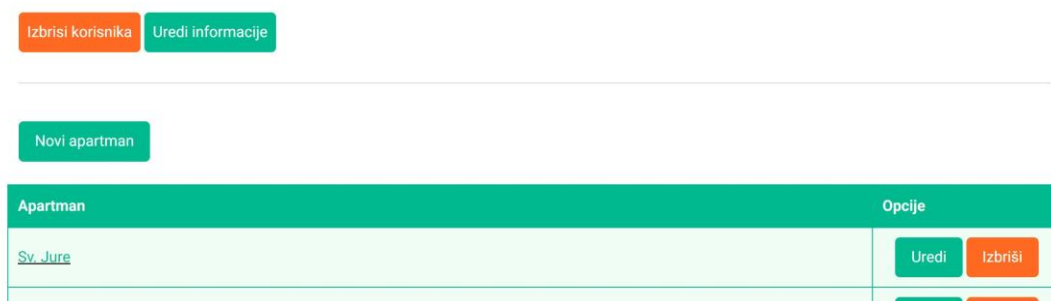
Ispis 25 – Označeni stupac tablice u Predlošku

Klikom na korisničko ime administrator može vidjeti detalje o tom korisniku. Također su dostupne opcije za urediti podatke korisnika i za njegovo brisanje. Korisnici (kao ni njihovi apartmani) nikada neće biti stvarno izbrisani iz baze podataka, već će biti označeni kao neaktivni, slično kao i kod apartmana. Ako administrator želi vidjeti sve izbrisane korisnike, može odabrati "Izbrisani" na vrhu stranice. Na aktivne korisnike se vraća klikom na "Aktivni".

Pogled izbrisanih korisnika je, kako bi se odmah razlikovao od aktivnih, obojan tamnom bojom. Neaktivnim korisnicima se mogu gledati detalji, ali ih nije moguće uređivati kako se statistika ne bi mogla manipulirati. No, dostupna je opcija za ponovno aktiviranje korisnika.

5.3 Detalji o korisniku i njihovo uređivanje

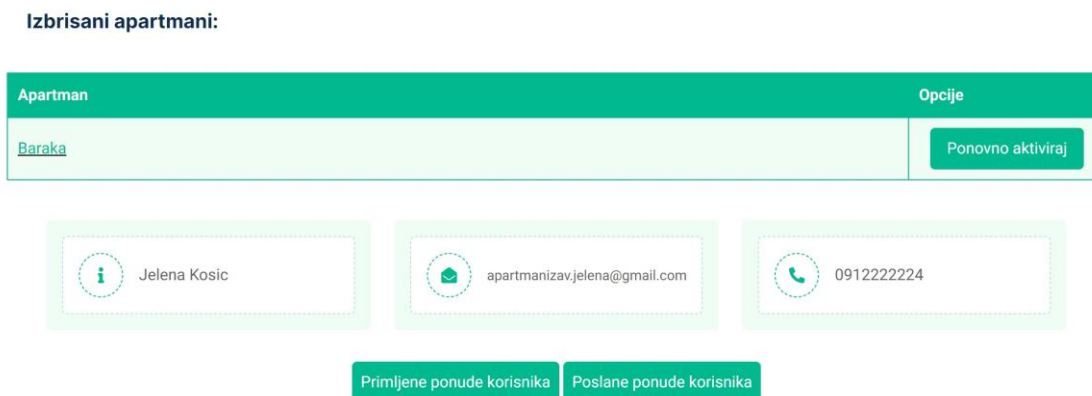
Na ovoj stranici su administratoru prikazani podaci o korisniku, kao i njegovi apartmani (slika 17). Pri ulasku na stranicu se, uz provjeru da je prijavljeni korisnik administrator, također provjerava da korisnik čiji su podaci prikazani nije i sam administrator, jer to nije dozvoljeno. Također se gleda je li on aktivan, jer je sučelje malo drugačije ovisno o tome. Ako nije, sučelje je tamnije kako bi to odmah bilo uočljivo. Također je dostupna opcija za ponovno aktiviranje korisnika, a u suprotnom su dostupne opcije za brisanje korisnika i njegovo uređivanje. Također, ako je aktivan, administrator korisniku može kreirati novi apartman.



Slika 17 - Aktivni apartmani od korisnika

Ovdje se mogu vidjeti i aktivni i izbrisani apartmani koji pripadaju korisniku. Kod aktivnih apartmana dostupne su opcije za uređivanje i brisanje apartmana, a kod izbrisanih apartmana oni se mogu ponovno aktivirati. U oba slučaja, može se kliknuti na naziv apartmana kako bi se došlo na njegovu stranicu. Administratorsko sučelje apartmana je već opisano u prvom poglavlju.

Nakon toga prikazane su informacije o korisniku koje su vidljive na njegovom profilu - ime i prezime, email i telefon. Na dnu su opcije koje vode na administratorsko sučelje korisnikovih primljenih i poslanih ponuda (slika 18).



Slika 18 - Izbrisani apartmani od korisnika

Administrator može korisniku urediti osnovne informacije sa profila. Kako bi to mogao, korisnik koji se uređuje ne smije biti administrator i mora biti aktivan. Forma za upisivanje podataka je jednaka kao i u korisničkom sučelju.

5.4 Brisanje korisnika

Administratoru je omogućena opcija brisanja korisnika. Korisnik će još uvijek ostati u bazi podataka, samo će mu biti dodana oznaka kako je neaktivan. Samo administrator može vidjeti korisnike koji su izbrisani. Administrator nema pravo brisati druge administratore.

Administrators će morati potvrditi odluku brisanja. Ako odluči izbrisati, nakon ponovnih provjera prava, napraviti će se kompletno brisanje korisnika i njegovih vezanih podataka. To znači, ako je korisnik imao apartmane, kako će se oni svi označiti kao neaktivni. Također, za svaki od apartmana, svi upiti koje je taj apartman zaprimio će također postati neaktivni. Osim toga, i svi upiti koje je korisnik poslao drugima će biti označeni neaktivnima (ispis 26). Ni jedno od ovih podataka neće nestati iz baze podataka, već se ovo radi kako ponude vezane za nešto što više ne postoji (apartman ili korisnika) ne

bi zbunjivale druge korisnike i kako se ne bi mogle raditi rezervacije za nepostojeće podatke.

```
if 'Izbrisi' in request.POST:
    if Apartmani.objects.filter(vlasnik = korisnik).exists():
        #svi apartmani korisnika
        ap_izbrisi = Apartmani.objects.filter(vlasnik = korisnik)
        for ap in ap_izbrisi:
            if Ponude.objects.filter(apartman = ap).exists():
                #sve ponude od njegovog apartmana
                pon_izbrisi = Ponude.objects.filter(apartman = ap)
                for pon in pon_izbrisi:
                    pon.aktivan = False
                    pon.save()
                ap.aktivan = False
                ap.save()
            if Ponude.objects.filter(gost = korisnik).exists():
                #sve ponude gdje je on gost
                pon_izbrisi2 = Ponude.objects.filter(gost = korisnik)
                for pon2 in pon_izbrisi2:
                    pon2.aktivan = False
                    pon2.save()
        korisnik.aktivan = False
        korisnik.save()
```

Ispis 26 – Brisanje apartmana

5.5 Stvaranje novog apartmana

Administrator nakon odabira nekog korisnika ima mogućnost njemu dodati novi apartman. Taj korisnik mora biti aktivan.

Apartman se dodaje na sličan način kao i u korisničkom sučelju, samo se za vlasnika apartmana ne postavlja prijavljeni korisnik, već korisnik kojega je administrator prethodno odabrao.

5.6 Ponovno aktiviranje izbrisanog apartmana

Osim brisanja apartmana, administrator ima moć i opet apartman vratiti kao aktivnog, bez obzira je li ga izbrisao on, neki drugi administrator ili sami njegov vlasnik.

Ako administrator odgovori potvrdno nakon što ga se upita je li siguran želi li to napraviti, osim što se apartman označava kao aktivan, također i svi upiti koje je apartman zaprimio postaju ponovno aktivni, jer su aktiviranje apartmana i korisnika jedini način za aktiviranje upita (analogno tome, brisanje apartmana i korisnika su jedini načini za njihovu deaktivaciju). To jest, aktivnost upita je samo odraz aktivnosti njemu vezanih podataka (ispis 27).

```
if 'Aktiviraj' in request.POST:
    if Ponude.objects.filter(apartman = apartman).exists():
        #sve ponude od apartmana
        pon_aktiviraj = Ponude.objects.filter(apartman = apartman)
        for pon in pon_aktiviraj:
            pon.aktivan = True
            pon.save()
        apartman.aktivan = True
        apartman.save()
```

Ispis 27 – Aktivacija apartmana

5.7 Ponovno aktiviranje izbrisanog korisnika

Administrator smije vratiti prethodno izbrisanog korisnika. Ta mogućnost isključivo pripada administratorskoj ulozi. Nije potrebno provjeriti ima li korisnik kojeg se vraća administratorsku ulogu, jer takav korisnik nikad i neće biti izbrisan.

Ako se pokrene postupak vraćanja izbrisanog korisnika, on će se označiti kao aktivan. Također će za svaki od korisnikovih apartmana aktivirati upiti poslani tom apartmanu, kao i upiti koje je korisnik slao drugima. To je zato što su, kad je korisnik bio izbrisan, svi ti upiti označeni neaktivnima. No, iako su i svi korisnikovi apartmani bili deaktivirani, oni se ovdje neće ponovno aktivirati, jer je moguće kako je korisnik neke od tih apartmana prethodno sam izbrisao.

5.8 Administratorsko sučelje pregleda upita

Kada administrator odabere nekog korisnika, može mu vidjeti sve upite koje je poslao i zaprimio.

Kod primljenih upita, može se vidjeti ime i prezime gosta, naziv apartmana na koji se može kliknuti kako bi se pogledale informacije o njemu, vrijeme potencijalnog noćenja te je li upit rezerviran, odbijen ili još neodgovoren. Administrator ima potpunu ovlast nad upitima, pa ih može rezervirati ili odbiti, ovisno o njihovom trenutnom statusu.

Administrator također može vidjeti i upite koji nisu aktivni, tj. koji pripadaju apartmanu ili korisniku koji je izbrisan, pomoću opejca na vrhu stranice. Administrator ih može samo pregledavati, a za aktivaciju mora ponovno aktivirati apartman ili korisnika s kojim su povezani. Njihovo sučelje je tamnije kako bi se razlikovali od aktivnih. Administrator može pogledati informacije o apartmanu i vidjeti je li upit bio rezerviran ili odbijen, no ne može mijenjati njegov status. I kod aktivnih i kod neaktivnih upita na vrhu stranice se klikom na korisnikovo ime mogu vidjeti podaci o njemu.

Sučelje poslanih upita je slično; može ih se rezervirati ili odbiti, osim ako su neaktivni, te otići na stranicu apartmana kojemu su poslani.

5.9 Brisanje recenzija

Administrator ima pravo izbrisati recenziju apartmana ako smatra kako je to potrebno. Recenzije su jedini podaci uz cijene koji se stvarno brišu iz baze podataka, umjesto označavanja kao neaktivni. Na administratorskom sučelju apartmana pokraj svake recenzije nalazi se opcija brisanja. Nakon provjere je li apartman aktivan, tj. ima li se pravo mijenjati njemu povezane podatke, administrator mora potvrditi želi li stvarno izbrisati recenziju. Ako odgovori potvrdno, obavlja se njeno brisanje iz baze podataka.

5.10 Popis svih apartmana

Administrator može vidjeti popis svih apartmana u aplikaciji. Može ih, slično kao kod popisa korisnika, filtrirati po bilo kojoj kombinaciji slijedećih parametara: naziv, grad, maksimalni broj osoba i broj zvjezdica (tj. kategorija). Popis gradova se uzima iz baze podataka, a zvjezdice se biraju iz padajućeg izbornika u vrijednostima od 1 do 5. Ne uzimaju se u obzir ako administrator ostavi "Izaberite ..." kao opciju. Za razliku od korisničke pretrage apartmana, gdje on upisuje koliko gostiju planira boraviti u apartmanu i gdje se zbog toga uključuju i apartmani sa većim kapacitetom od unesenog, ovdje administrator unosi točan maksimalni broj osoba koji apartman može primiti.

Apartmani se mogu poredati po bilo kojem stupcu kad se klikne na njegov naziv, tj. po nazivu, površini, vlasniku, gradu, kategoriji i kapacitetu. Kao i kod korisnika, stupac koji je odabran nije podcrtan. Klikom na naziv apartmana odlazi se na njegovu stranicu, a klikom na vlasnika na njegove informacije.

Može se birati između prikaza aktivnih i izbrisanih apartmana. Kod aktivnih, dostupne su opcije uređivanja i brisanja apartmana, a kod izbrisanih vraćanja apartmana. Kao i uvijek, neaktivni apartmani označeni su tamnijom bojom (slika 19).

Naziv: Grad: Maksimalni br. osoba: Broj zvijezdica:

Naziv	Vlasnik	Grad	Kategorija	Maksimalni br. osoba	Povrsina	Opcije
Baraka	Jelena Kosic	Split	4 Zvij.	4	50.23 m2	<input type="button" value="Ponovno aktiviraj"/>
Apartman Marta	Marta Lukic	Brela	3 Zvij.	2	50.00 m2	<input type="button" value="Ponovno aktiviraj"/>

Slika 19 - Izbrisani apartmani

6. Zaključak

Dobiveni rezultat ovog projekta je aplikacija koja korisnicima omogućava praćenje upita koje primaju njihovi apartmani te pretraživanje apartmana po kapacitetu, lokaciji, periodu zauzetosti i kategoriji. S druge strane, korisnici mogu slati upite drugim apartmanima te pratiti stanje njihovog upita. Dostupne su detaljne informacije o apartmanima i osnovni podaci o vlasniku. Sva komunikacija je automatizirana, tj. slanje upita i odgovaranje na njih odvija se preko biranja opcija na korisničkom sučelju i slanjem automatskih email poruka sa centralne email adrese aplikacije kako bi se obavijestilo korisnika o tom događaju. Za daljnju komunikaciju korisnici moraju samostalno poslati email poruku na adresu drugog korisnika. S tim na umu, aplikacija bi se mogla unaprijediti dodavanjem sustava privatnih poruka kako bi korisnici mogli međusobno komunicirati bez izlaska iz nje. Također, mogao bi se dodati sustav za obavijesti koji će korisnika direktno u aplikaciji obavještavati o događajima, npr. o primanju upita i gdje korisnici na centraliziranom mjestu mogu pregledati sve obavijesti koje su dobili.

Ova aplikacija rađena je u okviru *Django*. On je dosta olakšao proces njene izrade. Korisno je što su praktički sve potrebne komponente već ugrađene u samom okviru te se ne moraju instalirati, pa zbog toga ne nastaju problemi zbog kompatibilnosti jedne s drugom i u samoj instalaciji. Za dohvaćanje željenih podataka iz baze u Pogled koristi se maksimalno pojednostavljen način filtriranja podataka iz baze, ali je to ujedno i moćan alat za pisanje poslovne logike dobavljanja podataka bez potrebe za petljama. Vrlo je jednostavno i kreiranje tablica, posebno kod definiranja međusobnih veza između njih. Također je i dohvaćanje podataka preko tih veza lako i jasno te nije potrebno naglašavati kako, kada se uzimaju podaci iz neke tablice, također želi moći pristupiti i njoj vezanim podacima, već je to automatski ugrađeno. Programiranje u Predlošku je iznimno ograničeno. No, to ima svoje razloge, ponajviše kako se promjenom dizajna u Predlošku ne bi mogao slučajno narušiti sam program. Sve u svemu, *Django* se pokazao kao odlično okruženje, posebno za početnike, jer nudi mnoštvo opcija za prilagodbu, a sve komponente su već uključene pa se korisnik može koncentrirati samo na logiku svoje aplikacije.

7. Literatura

[1] A. Holovaty, J. Kaplan-Moss: “The Django Book,” Free Software Foundation Inc., Boston, SAD, 2002.

[2] SQLite, “About SQLite,” sqlite.org/about.html (posjećeno 31.5.2023.).

[3] Bootstrap, “Build fast, responsive sites with Bootstrap,” getbootstrap.com (posjećeno 31.5.2023.).