Implementacija aplikacije za prodaju i najam nekretnina u programskom jeziku Python

Počekal, Julija

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Julija Počekal

IMPLEMENTACIJA APLIKACIJE ZA PRODAJU I NAJAM NEKRETNINA U PROGRAMSKOM JEZIKU PYTHON

ZAVRŠNI RAD

Sisak, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Julija Počekal

Matični broj: 0115062255

Studij: ***Primjena informacijske tehnologije u poslovanju***

IMPLEMENTACIJA APLIKACIJE ZA PRODAJU I NAJAM NEKRETNINA U PROGRAMSKOM JEZIKU PYTHON

ZAVRŠNI RAD

Mentor:

Izv. prof. dr. sc. Markus Schatten

Sisak, rujan 2022

Julija Počekal

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

U ovom radu analizirat će se napravljena web aplikacija za prodaju i najam nekretnina u Hrvatskoj. Završni rad sastoji se od teorijskog i praktičnog dijela.

Teorijski dio obuhvaća opis korištenih tehnologija, opis strukture aplikacije i postavki sustava, opis implementacije i određenih interesantnih dijelova koda, prikaz aplikacije iz korisničke perspektive te sintezu rada s kritičkim osvrtom.

Praktični dio obuhvaća samu izradu aplikacije kroz Python programski jezik, Flask framework te HTML i CSS. Za izradu baze podataka korišteni su SQLite i Flask-SQLAlchemy. Aplikacija omogućava registraciju, prijavu i odjavu korisnika, stvaranje, brisanje i ažuriranje oglasa, filtriranje i pretraživanje postojećih oglasa te pregled vlastitog i tuđeg profila.

Ključne riječi: Python, Flask, SQLite, Flask-SQLAlchemy, web aplikacija

Sadržaj

1. Uvod

Tema završnog rada je implementacija web aplikacije za prodaju i najam nekretnina u Hrvatskoj kroz Python programski jezik. Aplikacija za praćenje oglasa nekretnina izvrstan je alat za sve koji se upuštaju u traženje nove nekretnine. Unutar aplikacije moguće je kreirati vlastiti oglas i pregledavati ostale dostupne oglase. Oglase je moguće pretraživati pomoću različitih kriterija i olakšati si potragu.

Interes za ovom temom javio mi se kada sam osobno bila u fazi traženja nekretnine te sam smatrala da će mi izrada ove stranice pomoći u učenju Python programskog jezika kao i boljem razumijevanju tržišta za nekretnine.

Današnja financijska situacija u Hrvatskoj povezana je sa sektorom nekretnina te stanovnici u Hrvatskoj svakodnevno zahtijevaju procjene vrijednosti nekretnina. Samim time radi se o konstantnom rastućem trendu potražnje za stanovima, kućama, poslovnim prostorima i slično. Kako i sam trend raste javlja se potreba i za izradom digitalnih aplikacija koje će pridonijeti ubrzanju i pojednostavljenju poslovnih procesa.

2. Metode i tehnike rada

Rad se sastoji od teorijskog i praktičnog dijela kroz koje se prati razvoj aplikacije. Kroz teorijski dio izrađuje se dokumentacija s opisivanjem ključnih riječi, analizom tehnologija, sustavnim praćenjem razvoja, primjerima korištenja aplikacije te kritičkog osvrta. Kao izvori rada koriste se članci i stručni radovi, dostupne knjige i službene web stranice. Za izradu dijagrama koristio se besplatni online alat Diagrams.net (https://www.diagrams.net/).

Prilikom izrade praktičnog dijela koristilo se integrirano razvojno okruženje PyCharm. Aplikacija je razvijena u Python programskom jeziku koristeći Flask framework. Kao stilski jezik korišten je CSS i framework Bootstrap, a HTML kao prezentacijski jezik za izradu web stranica. Za izradu baze podataka koristio se SQLite, a za olakšano pisanje upita koristio se Flask-SQLAlchemy.

3. Korištene tehnologije

Prilikom razvijanja ideje o izradi aplikacije odabrane su tehnologije i alati za korištenje koje su opisane u nastavku ovog rada.

3.1. PyCharm

Nguyen (2019., str. 8) navodi da je PyCharm integrirano razvojno okruženje (IDE) koje se koristi za programiranje u Python programskom jeziku. Prema njemu to je ujedno i najpopularnije razvojno okruženje za Python. Razvila ga je tvrtka JetBrains. PyCharm je razvijen od strane programera kako bi drugim programerima olakšali pisanje koda i pružili sve alate koji su potrebni za produktivni rad.

Prema JetBrains službenoj stranici, PyCharm nam olakšava rad na način da:

1. Analizira i dopunjuje kod
2. Provjera pogrešaka, podcrtava greške i nudi brze popravke
3. Ima jednostavnu navigaciju u projektu
4. Brine da je kod uredan
5. Nudi pomoć pri testiranju i izmjeni postojećeg koda
6. Omogućava prilagođavanje radnog prostora prema vlastitim preferencijama kao što su boja pozadine i funkcije tipki
7. Nudi odličnu podršku za različite framework-e kao što su Django, Flask, Pyramid i web2py
8. Ima mogućnost integracije s IPython Notebook i podržava Anacondu
9. Nudi više znanstvenih paketa kao što su matplotlib i NumPy
10. Sadrži veliku količinu alata kao što je ugrađeni terminal, rad s bazama podataka itd.
11. Podržava i druge jezike kao što su JavaScript, TypeScript, Cython, SQL, HTML, CSS, AngularJS, NodeJS i drugi
12. Omogućava rad na udaljenim hostovima ili virtualnim strojevima
13. Nudi podršku za najpopularnije sustave za verzioniranje kao što su Git, Mercurial i Subversion
14. Može se pokrenuti na više platformi kao što su Windows, MAC OS X i Linux.

S obzirom da uz profesionalnu verziju koja se plaća postoji i besplatna verzija, PyCharm je dostupan svima. Prilikom izrade ovog projekta koristila se PyCharm Community Edition 2020.3 x64 verzija. Ideja koja leži iza cijelog razvoja PyCharm-a je da se programeru omogući da više pažnje posveti samom programiranju, a da o ostalim tehnikalijama, kao što su uvlačenje i zagrade, brine razvojno okruženje. Odličan je alat za početnike jer ukazuje na propuste u sintaksi, nudi popis funkcija i paketa i samim time programeru štedi njegovo vrijeme.

3.2. Python

Službena Python stranica navodi da je Python programski jezik stvoren 1990. godine i jedan je od najbrže rastućih programskih jezika današnjice. Stvorio ga je Guido van Rossum, a samo ime je dobio po seriji Monty Python's Flying Circus. Prvi puta je objavljen 1991. godine.

Po samoj definiciji Python je objektno orijentirani programski jezik visoke razine koji se lako tumači i ima čitljivu sintaksu. Uz objektno orijentirani stil dopušta programerima i korištenje strukturnog i aspektno orijentiranog stila. Downey (2014., str.2) navodi da Python pripada grupi viših programskih jezika što znači da se brže piše, kraći je, lakše se čita i veća je vjerojatnost da će ispravno raditi. Također se može izvršavati na različitim vrstama računala uz neznatne ili minimalne izmjene. Iako se prvenstveno koristi kao skriptni jezik može se koristiti i za razvoj cjelovitih programa. Prema službenoj stranici Amazon Web Services (AWS) Python je interpretirani jezik što znači da se izvodi redak po redak. Ukoliko dođe do grešaka u kodu prestat će se izvoditi. Ono što Python čini odličnim jezikom je i njegova jednostavna sintaksa koja je slična engleskom jeziku, te ne koristi vitičaste zagrade već samo uvlačenja.

Prema Python službenoj stranici Python nam omogućava brži rad i učinkovitu integraciju sa sustavima. Odličan je jezik kako za početnike tako i profesionalce.

Prednosti Python-a uključuju:

1. Programeri mogu lako razumjeti i čitati kod jer ima jednostavnu sintaksu
2. Program se može napisati koristeći manje redaka u usporedbi s nekim drugim jezicima
3. U Pythonu se ne moraju deklarirati tipovi varijabli

4. Posjeduje veliku standardnu biblioteku koja sadrži kodove za višekratnu upotrebu za gotovo sve zadatke
5. Može se jednostavno koristiti s drugim programskim jezicima
6. Prenosiv je na različite operativne sustave
7. Python zajednica je jako velika i pruža odličnu podršku svim programerima
8. Puno korisnih izvora dostupno je na internetu

Python ima i više slučajeva upotrebe u razvoju aplikacija. Neke od njih su:

1. **Web razvoj na strani poslužitelja** – uključuje složene pozadinske funkcije potrebne za prikazivanje informacija korisniku na web stranicama (komuniciranje s bazom podataka i drugim web mjestima, zaštita podataka). Python pruža više framework-a za bržu izradu web aplikacija.
2. **Automatizacija s Python skriptama** – koristi se za automatizaciju svakodnevnih zadataka kao što su preimenovanje velikog broja datoteka odjednom, pretvaranje datoteke u drugu vrstu, uklanjanje dvostrukih riječi, izvođenje matematičkih operacija, slanje e-mail poruka, preuzimanje sadržaja, pronalaženje pogrešaka u datoteci i drugo.
3. **Podatkovna znanost i strojno učenje** – izvlači vrijedno znanje iz podataka i uči računala da automatski uče iz podataka i prave točna predviđanja. Python se koristi za ispravljanje i uklanjanje netočnih podataka, ekstrakciju i odabir različitih značajki podataka, označavanje podataka, statistika podataka, vizualizacija podataka
4. **Razvoj softvera** - praćenje grešaka u softverskom kodu, automatska izgradnja softvera, razvoj softverskih prototipova, razvoj složenih i jednostavnijih videoigara i drugo.
5. **Automatizacija testiranja softvera** – provjera podudaranja stvarnih rezultata softvera s očekivanim rezultatima i pisanje testnih slučajeva. Programeri koriste jedinično testiranje kao što je Unittest, Robot i PyUnit.

Python je odličan jezik kako za male programe tako i za velike i složene. Čist je i uredan jezik stoga se i najvećim projektima lako upravlja. Dostupan je svima, a zbog svoje jednostavnosti rješavanje problema izgleda lako kao da zapisujemo svoje misli o rješenju. Za izradu ovog projekta koristio se Python 3.8.

3.3. Flask

Framework ili okvir je skup biblioteka i modula koji se koriste za brži razvoj Python aplikacije. Flask je web framework koji je napisan u Python-u. Real Python stranica navodi kako je Python razvio Armin Ronacher. To je modul koji omogućava jednostavan razvoj web aplikacija jer programeri ne moraju brinuti o detaljima kao što su na primjer protokoli. Flask spada u kategoriju mikrookvira jer pruža minimalne funkcionalnosti za izgradnju jednostavnih aplikacija i ne zahtjeva posebne biblioteke i alate. Prema The Pallets Projects službenoj stranici Flask je osmišljen kako bi početak razvoja aplikacije bio brz i jednostavan uz mogućnost povećanja aplikacije u one složenije. Flask nudi prijedloge ali ne nameće određeni izgled aplikacije ili zavisnosti o drugim alatima stoga možemo reći da ima malu jezgru koja se može proširiti. Na programeru je da izabere alate i biblioteke koje želi koristiti.

Flask se temelji na određenim komponentama:

1. **Web Server Gateway Interface (WSGI)** – pristupno sučelje web poslužitelja koje se koristi kao standard za razvoj Python web aplikacija
2. **Werkzeug** – biblioteka pomoćnih programa za WSGI alat koji implementira zahtjeve, odgovore i funkcije.
3. **Jinja2** – mehanizam za pisanje predložaka. Osigurava posebna mjesta u predlošku za pisanje koda sličnog sintaksi Python-a

S obzirom na postojanje drugih framework-a mnogi se pitaju zašto odabrati baš Flask. Flask je za razliku od framework-a Django vrlo pythonic što znači da iskorištava jedinstvene značajke Python programskog jezika. Python Basics stranica navodi kako je Flask vrlo eksplicitan jezik što povećava čitljivost i još je jedna od prednost tog framework-a.

Grinberg (2014.) navodi kako je Flask dovoljno mali framework da kada ga osoba jednom upozna može lako čitati njegov izvorni kod, no iako je mali ne znači da radi manje od drugih. Flask je vrlo popularan što znači da je ažuriran i moderan. U početku je bio jednostavan framework, a sada je postao jedan od najpopularnijih Python framework-a. Njegova funkcionalnost se lako proširuje, te se može proširiti na veće aplikacije.

3.4. SQLite

Prema Haldaru (2007., str. 1) SQLite je mali sustav za upravljanje relacijskom bazom podataka, bez konfiguracije i potrebnog servera, prilagođen korisniku, lako se održava, orijentiran na transakciju i baziran na SQL-u. Temeljen je na maloj C programskoj biblioteci koja je samostojeća i ne ovisi o drugim komponentama. Dizajnirao ju je D. Richard Hipp 2000. godine stoga se smatra relativno mladim sustavom za upravljanje bazom podataka (SUBP). Što se tiče popularnosti SQLite je jedan od sustava s najbržim porastom. SQLite kod je javno dostupan i besplatan. Iako se češće koristi u manjim projektima, može se koristiti i u onim složenijim i većim. SQLite je ugrađen u sve mobilne telefone i većinu računala, te dolazi u paketu s velikim brojem drugih aplikacija koje se koriste na dnevnoj bazi.

Osim već navedenih, prednosti SQLite sustava za upravljanje bazom podataka prema TutorialsPoint stranici su:

1. SQLite dolazi s nultom konfiguracijom što znači da nije potrebno postavljanje niti administracije
2. Cijela SQLite baza podataka pohranjena je u jednu datoteku na disku
3. Vrlo je malen i lagan
4. Samostalan je što znači da nema vanjskih ovisnosti
5. SQLite transakcije su usklađene s ACID-om što osigurava da podaci ostaju postojani prilikom pada sustava
6. Podržava većinu značajki koje se nalaze u standardu SQL92
7. Podržava jednostavan API koji je lak za korištenje
8. Dostupan je na Windows i Unix sustavima
9. Posjeduje brzu izvedbu standardnih operacija
10. Podaci iz baze se slobodno mogu dijeliti između više računala

Iako ima veliki broj prednosti, SQLite ipak ima i neke nedostatke. Podaci iz baze se mogu dijeliti između više računala što je prednost, no ukoliko imamo veliki broj klijentskih programa koji šalju upite istoj bazi podataka putem iste mreže tada može doći do značajnog kašnjenja. Također bih se SQLite trebao manje koristiti kod web stranica s velikim prometom. Zbog čestog zapisivanja podataka u bazu trebao bi se koristiti profesionalniji sustav za upravljanje bazom podataka. S obzirom da su podaci dostupni na jednom mjestu, također postoji rizik krađe ili gubitka podataka.

Čak i ako uzmemo u obzir nedostatke, prednosti mogućnosti pristupa i manipuliranja bazom podataka bez uključivanja poslužiteljske aplikacije su ogromne. Codecademy stranica navodi kako se SQLite koristi diljem svijeta za testiranje, razvoj i u bilo kojem drugom scenariju gdje ima smisla da baza podataka bude na istom disku kao i kod aplikacije. Održavatelji SQLite-a smatraju da je to jedan od najrepliciranijih dijelova softvera na svijetu. Ono što je također bitno za našu aplikaciju je da SQLite dolazi s Flask-om.

3.5. Flask-SQLAlchemy

Prema SQLAlchemy službenoj dokumentaciji SQLAlchemy je SQL alat otvorenog koda i objektno-relacijski maper (ORM) za programski jezik Python. SQLAlchemy programerima omogućava fleksibilnost pisanja SQL koda. Pruža pristup bazama podataka na pythonic način. Umjesto klasičnih SQL upita, SQLAlchemy ima više oblik samog Python programskog jezika. Iako umjesto SQL sintakse pišemo Python, ORM nam pomaže prebaciti sav kod u SQL naredbe.

Flask-SQLAlchemy je proširenje za Flask koji aplikaciji omogućuje jednostavno povezivanje Flask-a i SQLAlchemy-ja. Prema službenoj dokumentaciji cilj mu je pojednostaviti korištenje SQLAlchemy-ja s Flask-om. Pruža dodatne zadane postavke i pomoćnike koji olakšavaju izvršavanje uobičajenih zadataka.

Flask SQLAlchemy je ORM alat koji uspostavlja odnos između objekata i tablica relacijskih baza podataka. Službena dokumentacija navodi da je preslikavanje između oba važno jer je Python sposoban pohranjivati podatke u obliku objekata, dok baza podataka pohranjuje podatke u obliku relacijskih tablica, tj. zbirke redaka i stupaca. Objektno-relacijsko preslikavanje tehnika je pohranjivanja python objekata u tablice baze podataka bez pisanja neobrađenih SQL upita.

3.6. CSS i Bootstrap

Cvetković i sur. (2010., str. 241) definiraju CSS (Cascading Style Sheets) kao specifikaciju koja služi za definiranje stilova koji određuju izgled HTML elementa. Ti stilovi kao što su boja, font, pozicija, dimenzija i dr. se čuvaju u posebnoj datoteci s ekstenzijom .css ili su interni pa se vežu za jednu stranicu kojoj pripadaju. Također mogu biti i inline tj. u okviru elementa. Web dokument sastoji se od tri dijela: sadržaj, prezentacija i ponašanje. CSS je jezik stila koji se brine o prezentacijskom dijelu aplikacije. On oblikuje sadržaj dokumenta koji je napisan HTML jezikom. Definira kako prikazati HTML elemente i uređuje se sam izgled i raspored stranice. Cvetković i sur. (2010., str. 242-243) govore kako CSS dizajnerima pruža dvije ključne stvari, a to su precizno razdvajanje forme od sadržaja, te kontrolu velike količine dokumenata. Stilovi omogućuju veću kontrolu uz pisanje manje koda što ubrzava proces izrade i održavanja stranice.

Isto kao što Python ima različite framework-e ima ih i CSS. Bootstrap službena stranica navodi kako je Bootstrap biblioteka koja omogućava lakši dizajn. Najpopularniji je framework CSS-a. To je framework otvorenog koda i usmjeren je na responzivni dizajn. Sačinjen je od HTML-a, CSS-a i (po izboru) JavaScript-a. Fokusiran je na pojednostavljivanje razvoja web stranica. Sastoji se od raznih predložaka i komponenti s uključenim stilom. Korištenjem Bootstrap-a dobivamo ujednačen izgled elemenata, tablica, obrazaca u svim web preglednicima. To je snažan i proširiv set alata za sučelje prepun značajki. Bootstrap službena stranica navodi kako je njegova najistaknutija komponenta tzv. kontejner budući da je većina elemenata smještena unutar njega. Komponente poput gumba, formi i obrazaca, navigacije, tablica i dr. vrlo se jednostavno koriste. Potrebno je gotovo samo kopirati kod s Bootstrap stranice i prebaciti ga u svoj kod. Bootstrap vrlo jednostavno omogućuje prilagodbu stranice svim veličinama ekrana, a svojim već određenim stilovima odiše jednostavnošću, ujednačenošću i modernim dizajnom.

3.7. HTML

HTML (HyperText Markup Language) Cvetković i sur. (2010., str. 25) definiraju kao opisni jezik specijalno namijenjen web stranicama. Prva verzija HTML-a objavljena je 1991. godine i stvorio ga je Tim Berners-Lee. To je jezik za označavanje hiperteksta i pomoću njega jednostavno se mogu odvojiti elementi kao što su naslov, paragrafi, linkovi, citati i slično. Unutar HTML-a su ugrađeni i elementi koji detaljnije opisuju sam dokumenti, a poznati su kao meta podaci.

Cvetković i sur. (2010., str.25) navode kako se HTML jezikom oblikuje sadržaj i stvaraju se hiperveze. Smatraju kako je HTML u početku bio ograničen što se označavanja sadržaja tiče i pružao je samo elementarne stvari za formatiranje teksta. Kako je sam Web rastao došlo je do većih potreba za bogatijim sadržajem te se razvio HTML standard. Osnovu HTML-a predstavljaju tag-ovi i atributi. Pomoću tag-ova odvaja se određeni dio dokumenta od drugih i na njega se primjenjuju pravila definirana samim tag-om. Atributi se nalaze unutar samih tag-ova i omogućavaju da se pokraj imena tag-a još bliže odredi način ponašanja i prikaza. Prema MDN službenoj stranici „hipertekst“ se odnosi na veze koje povezuju web stranice jednu s drugom. Veze su temeljni aspekt web-a, te samim postavljanjem sadržaja na Internet i njegovim povezivanjem sa stranicama koje su izradili drugi ljudi, postajemo aktivni sudionik Word Wide Web-a.

HTML je vrlo jednostavan i lagano se uči što je i razlog za njegovu sveopću prihvaćenost. On je temelj svih web stranica i bez njega ne bismo mogli organizirati tekst ili dodati slike i videozapise na svoje web stranice. HTML se može smatrati početkom onoga što se mora znati kako bismo mogli raditi web stranice.

.

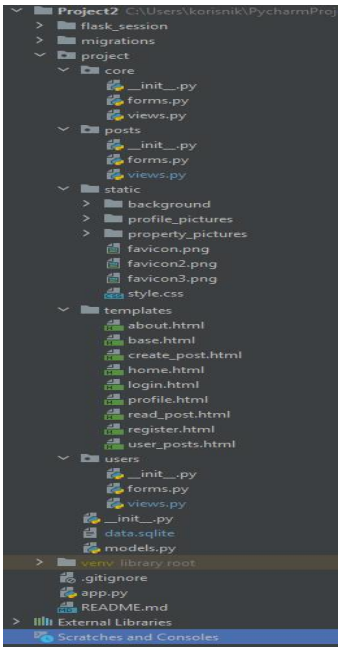
4. Implementacija

Unutar poglavlja pobliže će se opisati sama izrada projekta, struktura aplikacije i način na koji su se tehnologije i alati primijenili.

4.1. Struktura aplikacije

Projekt je sastavljen od više direktorija koji ponovo sadrže poddirektorije i dokumente. Glavni direktorij je Project2. Unutar njega postoje direktoriji:

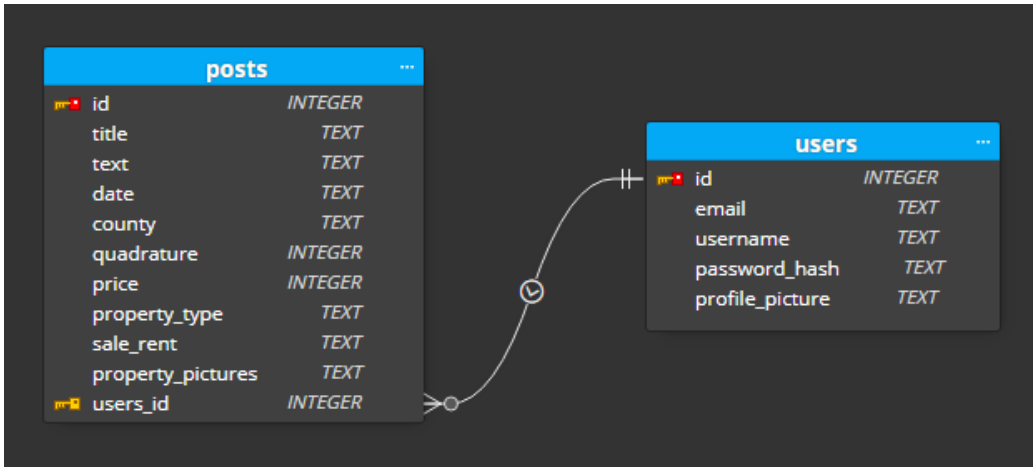
- **flask_session** – direktorij u kojem su pohranjene datoteke sesije
- **migrations** – direktorij u kojem su pohranjene datoteke migracija
- **project** – direktorij koji sadrži svu logiku i izgled projekta i dijeli se na poddirektorije core (forme i pogledi za početnu stranicu i temelje aplikacije), posts (forme i pogledi vezani za postove), static (sadrži dokument za stil, spremljene slika), templates (predlošci), users (forme i pogledi vezani za korisnike), __init__.py (konfiguracije), data.sqlite (baza podataka), models.py (kreiranje modela), app.py (pokretanje aplikacije)
- **venv** – direktorij u kojem su pohranjene datoteke virtualnog okruženja i instaliranih paketa



Slika 1 Struktura projekta

4.1.1. ERA model baze podataka

Prilikom izrade projekta jedna od prvih stavki je izrada ERA modela baze podataka. To je model koji prikazuje odnose između entiteta i atributa unutar baze podataka. S obzirom da je ova aplikacija napravljena uz samo dva entiteta, ERA model je prilično jednostavan.



Slika 2 ERA model baze podataka

Entiteti posts i users imaju id kao primarne ključeve dok posts ima i vanjski ključ users_id. Preko vanjskog ključa napravljena je vezu između entiteta. Povezani su preko veze jedan obavezni prema više opcionalnih što znači da jedan user može imati jedan ili više postova ili uopće ne mora. Entitet posts ima i ostale atribute title, text, date, county, quadrature, price, property_type, sale_rent i property_pictures, a users ima atribute email, username, password_hash i profile_picture.

4.1.2. Kreiranje baze podataka

Nakon napravljenog ERA modela kreirana je baza podataka. S obzirom da je korišten Flask-SQLAlchemy kod izgleda više kao Python, a manje kao SQL. Unutar dokumenta *models.py* napravljene su tablice i stupci.

```
class Posts(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey("users.id"),
    nullable=False)
    title = db.Column(db.String(124), nullable=False)
    text = db.Column(db.Text, nullable=False)
    date = db.Column(db.DateTime, nullable=False,
    default=datetime.utcnow)
    county = db.Column(db.String(124), nullable=False)
    quadrature = db.Column(db.Integer, nullable=False)
    price = db.Column(db.Integer, nullable=False)
    property_type = db.Column(db.String(124), nullable=False)
    sale_rent = db.Column(db.String(124), nullable=False)
    property_pictures = db.Column(db.String(1000),
    default="default_property.png")

    def __init__(self, user_id, title, text, county, quadrature,
    price, property_type, sale_rent, property_pictures):
        self.user_id = user_id
        self.title = title
        self.text = text
        self.county = county
        self.quadrature = quadrature
        self.price = price
        self.property_type = property_type
        self.sale_rent = sale_rent
        self.property_pictures = property_pictures
```

U priloženom kodu napravljena je tablica pod imenom posts. Tablica posts sadrži stupce id, user_id, title, text, county, quadrature, price, property_type, sale_rent i property_pictures. Id je ujedno i primarni ključ dok je user_id vanjski ključ i poveznica s drugom tablicom. Također su za svaki stupac definirani tipovi podataka i neka druga svojstva.

Nakon definiranih stupaca korišten je konstruktor __init__ kako bi se inicijaliziralo stanje objekta. Time se dodjeljuje vrijednost podacima klase prilikom samog kreiranja klase.

```
class Users(db.Model, UserMixin):
      id = db.Column(db.Integer, primary_key=True)
      email = db.Column(db.String(124), unique=True, nullable=False)
      username = db.Column(db.String(124), unique=True,
      nullable=False)
      password_hash = db.Column(db.String(164), nullable=False)
      profile_picture = db.Column(db.String(124),
      default="default.jpg")
      posts = db.relationship("Posts", backref="author",
      lazy="dynamic")

      def __init__(self, email, username, password):
            self.email = email
            self.username = username
            self.password_hash = generate_password_hash(password)

      def check_password(self, password):
            return check_password_hash(self.password_hash, password)
```

Tablica users sastoji se od stupaca id, email, username, password_hash, profile_picture i posts. Stupac id je ujedno i primarni ključ. Stupac posts spaja korisnika s njegovim objavama, a veza između modela Posts i Users definirana je pomoću backref-a. Pomoću konstruktora __init__ inicijalizirano je stanje objekta. Zbog sigurnosti ne pohranjujemo sirovi tekst kao lozinku već se koristi hashing. Biblioteka za hashing koja je korištena je Werkzeug. Ona sadrži check_password_hash funkciju. Kako bismo provjerili točnost lozinke korisnika dodana je funkcija check_password. Njoj prosljeđujemo lozinku prilikom prijave te se ona uspoređuje s hashiranom lozinkom iz baze podataka pomoću check_password_hash funkcije.

4.2. Postavke sustava

Na samom početku izrade projekta potrebno je instalirati određene pakete, povezati ih s aplikacijom i stvoriti virtualno okruženje. S obzirom da je korišten PyCharm virtualno okruženje je napravljeno automatski kada je kreiran projekt. Unutar virtualnog okruženja potrebno je instalirati dodatne pakete koji će se koristiti. Koristeći pip (instalacijski program za Python) instalirani su:

- **Flask** – framework za izradu web stranica
- **Flask-SQLAlchemy** – proširenje koje Flask-u dodaje podršku za SQLAlchemy
- **Flask-Migrate** – upravlja migracijama nad bazom podataka, osigurava da se sve napravljene promjene unutar modela primijene
- **Flask-WTF** – za prikazivanje i provjeru obrazaca na siguran i fleksibilan način
- **Flask-Session** – dodaje podršku za sesiju na strani poslužitelja
- **Flask-Login** – upravljanje korisničkim sesijama kao što su prijava i odjava
- **Pillow** – upravljanje, obrada slika, podrška za formate datoteka
- **email-validator** – provjera ispravnosti email adrese unutar obrazaca

Nakon instalacije paketa potrebno je odraditi određene postavke. Kreirani su potrebni direktoriji i dokumenti koji su prikazani u poglavlju „Struktura aplikacije“ (Slika 1). Unutar __init__.py dokumenta uvezeni su instalirani paketi i odrađene su potrebne konfiguracije:

1. Stvaranje aplikacije

```
app = Flask(__name__)
```

2. Postavljanje tajnog ključa kako bismo mogli kreirati forme

```
app.config["SECRET_KEY"] = "secretkey"
```

3. Postavljanje glavnog direktorija gdje se projekt nalazi

```
basedir = os.path.abspath(os.path.dirname(__file__))
```

4. Postavljanje lokacije baze podataka

```
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///" +
os.path.join(basedir, "data.sqlite")
```

5. Onemogućeno praćenje svih promjena na bazi podataka

```
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
```

6. Postavljanje sesije

```
SESSION_TYPE = 'filesystem'
app.config.from_object(__name__)
Session(app)
```

7. Povezivanje aplikacije i SQLAlchemy

```
db = SQLAlchemy(app)
```

8. Povezivanje aplikacije s bazom podataka

```
Migrate(app, db)
```

9. Stvaranje LoginManager klase i povezivanje s aplikacijom

```
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = "users.login"
```

U app.py dokument je uvezena aplikacija koja se pokreće iz istog dokumenta pomoću koda:

```
if __name__ == "__main__":
    app.run(debug=True)
```

Nakon što je cijela konfiguracija postavljena, u terminalu je određen način učitavanja aplikacija:

```
set FLASK_APP = app.py
```

Kako bi se moglo raditi s migracijama, u terminalu je postavljen direktorij migracija i migracijski dokument, a potom je nadograđena baza podataka s migracijama

```
flask db init
flask db migrate –m „first migration“
flask db upgrade
```

4.3. Interesantni dijelovi koda

Unutar ovog poglavlja izdvojeni su dijelovi koda koji su mi kao početniku bili novi i nešto zahtjevniji.

4.3.1. Funkcija za obradu i spremanje slika

Aplikacija za nekretnine zahtjeva veliki broj slika. Kako bi se slike pohranjivale bilo je potrebno napraviti funkciju koja će ujedno obraditi sliku i spremiti ju unutar projekta. Unutar funkcije koristili su se moduli:

- **secrets** – koristi se za generiranje kriptografskih jakih nasumičnih brojeva
- **os** – pruža funkcije za interakciju s operativnim sustavom
- **pillow** – za obradu i spremanje slika

```
def save_picture(form_picture):
      random_hex = secrets.token_hex(8)
      _, ext = os.path.splitext(form_picture.filename)
      picture_filename = random_hex + ext
      picture_path = os.path.join(basedir, "static/profile_pictures",
      picture_filename)
      picture_size = (200, 200)
      i = Image.open(form_picture)
      i.thumbnail(picture_size)
      i.save(picture_path)
      return picture_filename
```

Funkcija pod imenom save_picture prihvaća jedan parametar form_picture koji će se proslijediti iz forme. Unutar funkcije, izvlači se ime slike i odvaja od ekstenzije da bi umjesto imena stavili nasumično generirani broj kako bi imena unutar baze bila jedinstvena. Koristeći

modul os definira se gdje će se unutar projekta spremiti slika. S modulom pillow učitava se slika kako bi se mogla obraditi. Obrađuje se veličina slike te se sprema na već određeno mjesto spremanja. Na kraju funkcija vraća novo ime slike zajedno s ekstenzijom.

4.3.2. Blueprints

Prilikom strukturiranja aplikacije, blueprint može uvelike pomoći. Prema definiciji s RealPython stranice, blueprint predstavlja nacrt kako konstruirati ili proširiti aplikaciju. Njime se mogu grupirati određene funkcionalnosti u komponente koje se mogu ponovo koristiti. Prilikom stvaranja mogućnosti za prijavu i odjavu korisnika, unutar direktorija users stvoren je dokument *views.py*. Kako bi se blueprint stvorio prvo ga moramo uvesti putem import-a, a potrebno ga je i definirati unutar tog dokumenta da bi se mogao koristiti s pogledima (routes).

```
users = Blueprint("users", __name__)
```

Nakon što smo ga definirali potrebno je modificirati *__init__.py* dokument i registrirati blueprint kako bi se mogao koristiti.

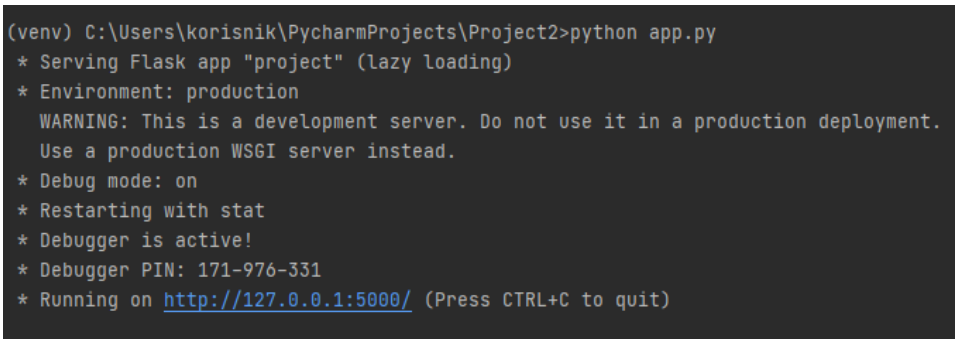
```
from project.users.views import users

app.register_blueprint(users)
```

Time je blueprint spreman za korištenje. Unutar aplikacije također su napravljeni blueprint-i i za objave (posts) i bazu stranice (core).

5. Primjer korištenja

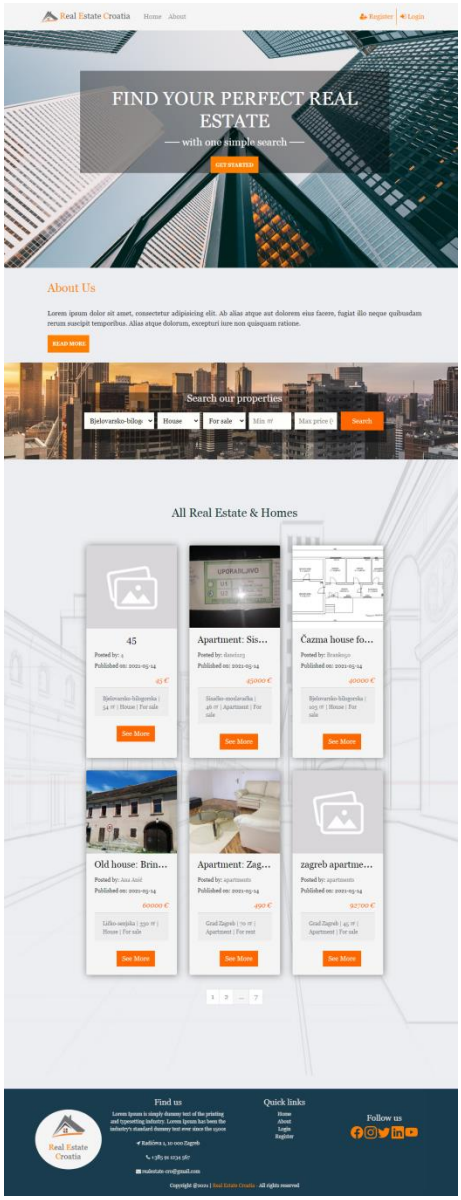
Kako bismo aplikaciju mogli koristiti potrebno ju je prvo pokrenuti. Unutar PyCharm-a dolazi ugrađeni terminal koji se ujedno koristi za pokretanje aplikacije. U terminalu upišemo naredbu `python app.py` i aplikacija se pokreće na adresi http://127.0.0.1:5000/.



Slika 3 Pokretanje aplikacije u terminalu

IP adresa 127.0.0.1 je IPv4 adresa posebne namjene i naziva se lokalna adresa (*localhost*). Prema PureVPN službenoj stranici sva računala koriste tu adresu kao svoju, ali ona ne dopušta računalima da komuniciraju s drugim uređajima kao što to čini prava IP adresa. Poruke poslane na lokalne ili povratne adrese ne dopiru izvan lokalne mreže. Broj priključka (u ovom slučaju 5000) je način za identifikaciju određenog procesa kojem će se neka poruka proslijediti kada stigne na poslužitelj.

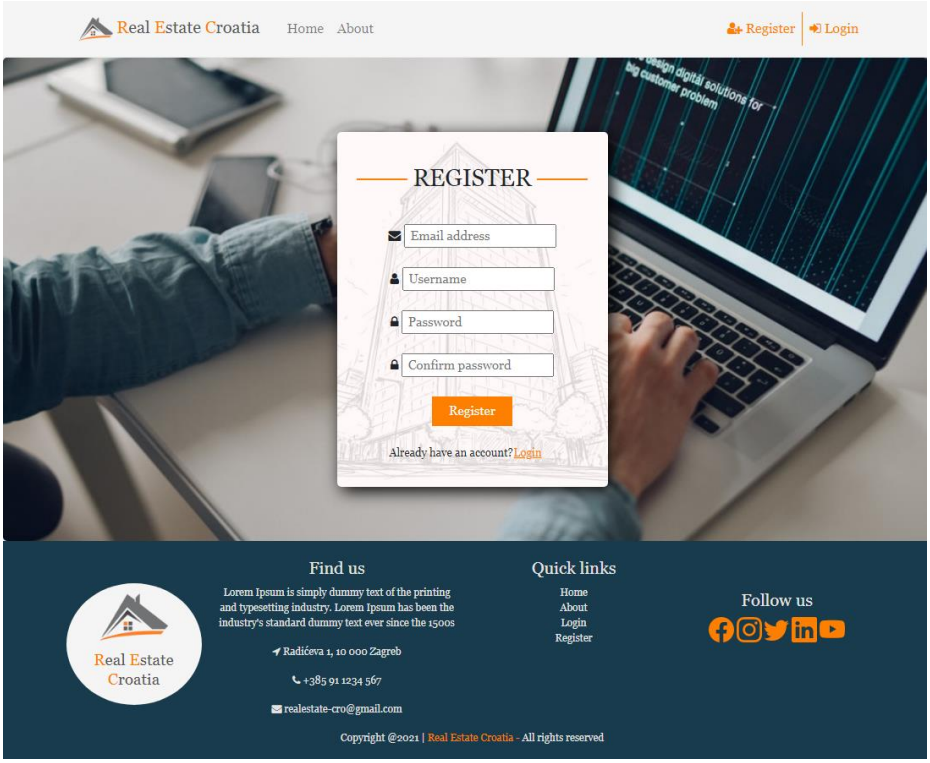
Nakon što otvorimo adresu aplikacija se pokreće i prikazuje se početna stranica aplikacije.



Slika 4 Početna stranica

5.1. Registracija i prijava korisnika

Kako bi se korisnik prijavio na stranicu mora se prvo registrirati (ukoliko to već nije učinio). Na navigacijskoj traci izabire opciju *Register* i otvara mu se forma koju mora popuniti.

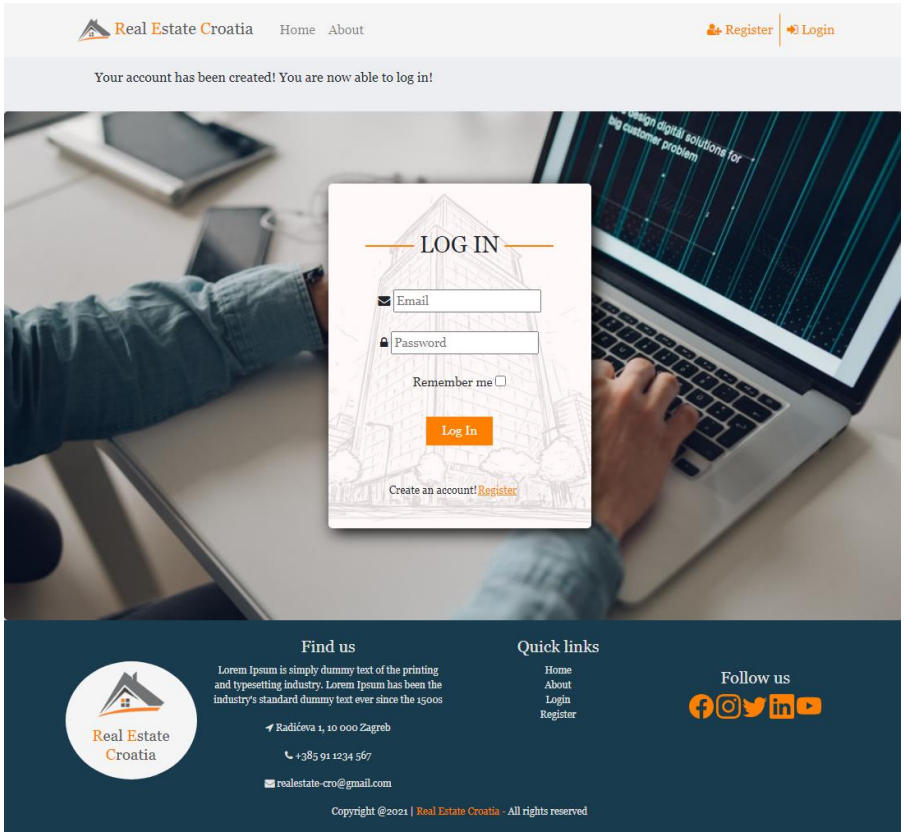


Slika 5 Registracija korisnika

Ukoliko korisnik upiše krive podatke, unutar forme se izbacuju upozorenja kao što je prikazano na primjeru na slici 6.

Slika 6 Neuspješna registracija

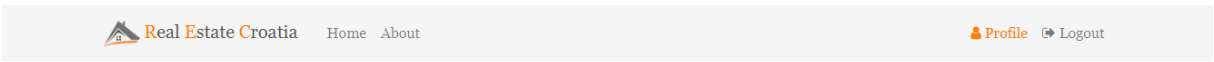
Nakon uspješne registracije korisnik je preusmjeren na prijavu.



Slika 7 Prijava korisnika

Kada se korisnik uspješno prijavi, preusmjeren je na početnu stranicu i tada ima mogućnosti uređivanja svojeg profila, informacija, stvaranje oglasa, pregledavanje već postojećih oglasa i slično.

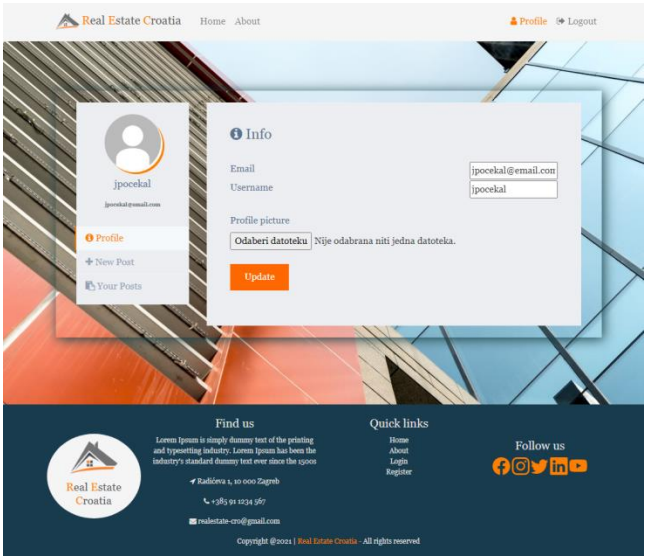
Ukoliko se prijavljeni korisnik želi odjaviti potrebno je samo pritisnuti na *Logout* u navigacijskoj traci.



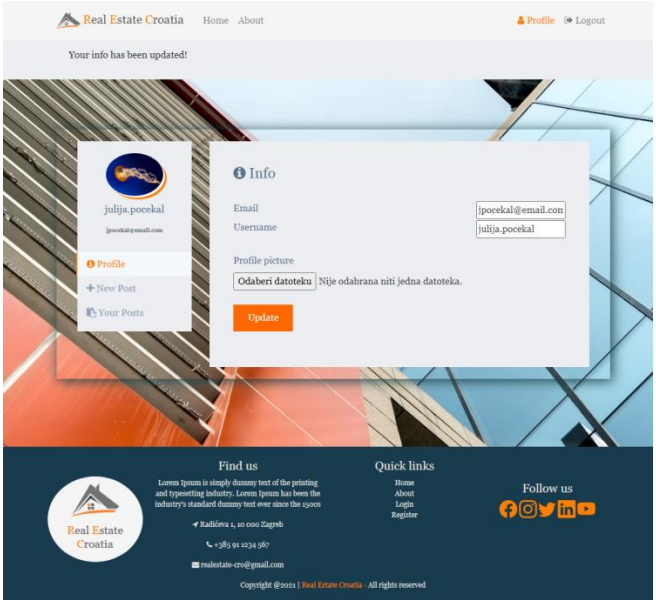
Slika 8 Navigacijska traka prijavljenog korisnika

5.2. Ažuriranje informacija korisnika

Registrirani korisnik ima mogućnost odlaska na svoj profil gdje su prikazane informacije o njemu. Korisnik može promijeniti svoju email adresu i korisničko ime, te može promijeniti svoju sliku profila. Ukoliko korisnik želi ažurirati svoje podatke potrebno je na navigacijskoj traci pritisnuti na *Profile*. Otvorit će se profilna stranica korisnika gdje može napraviti izmjene, a pritiskom na gumb *Update* pohranit će se nove informacije.

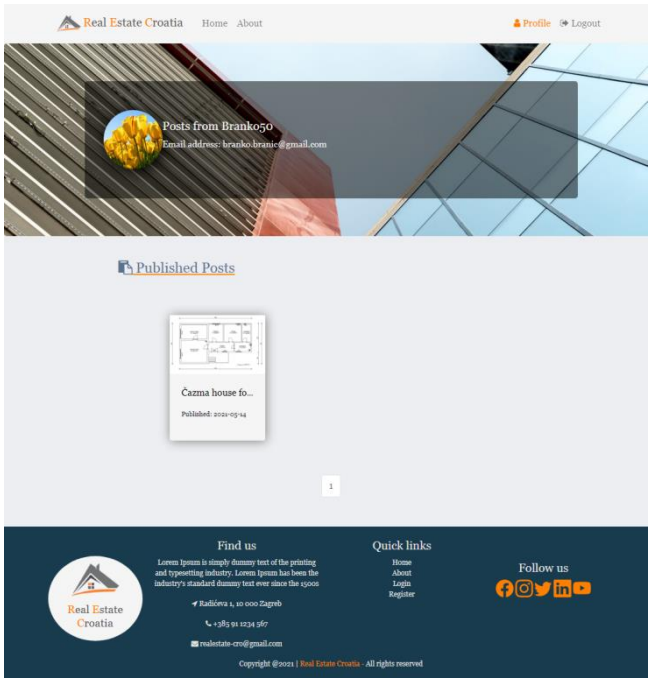


Slika 9 Profil korisnika



Slika 10 Ažurirani podaci korisnika

Svi korisnici (registrirani i neregistrirani) imaju mogućnost pregleda informacija o osobi koja je objavila oglas. Kada pritisnu na ime korisnika unutar njegovog oglasa prikazat će se stranica s njegovim informacijama i svim objavljenim oglasima.



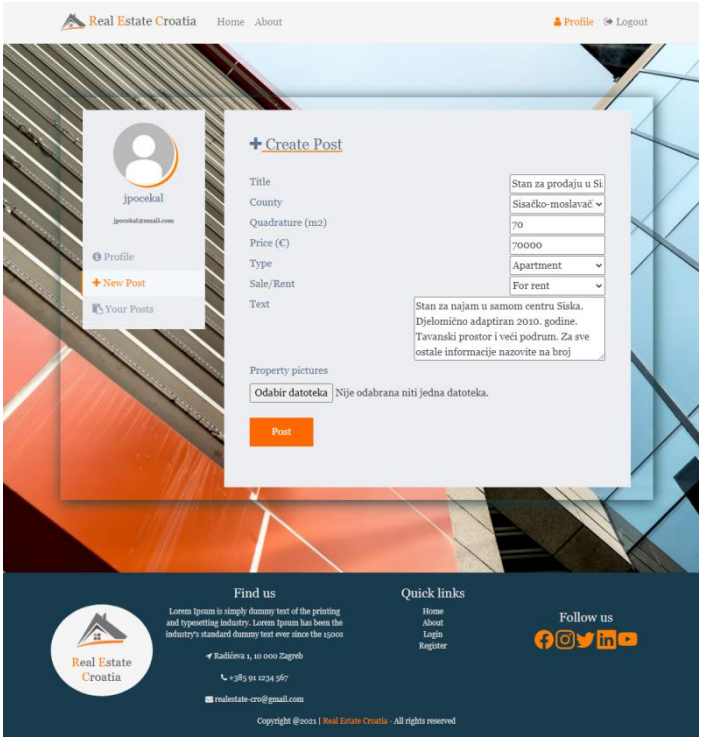
Slika 11 Informacije o korisniku

5.3. Mogućnosti oglasa

S obzirom da se aplikacija temelji na objavljivanju i pregledavanju oglasa vezanih za nekretnine u Hrvatskoj bitna značajka je da i neregistrirani korisnici mogu pregledavati oglase, no oni ne mogu stvarati svoje oglase. Ukoliko žele objaviti oglas potrebno se prije toga registrirati. Kada se korisnik prijavi tada dobiva mogućnost kreiranja oglasa, pregleda, ažuriranja i brisanja svojih oglasa.

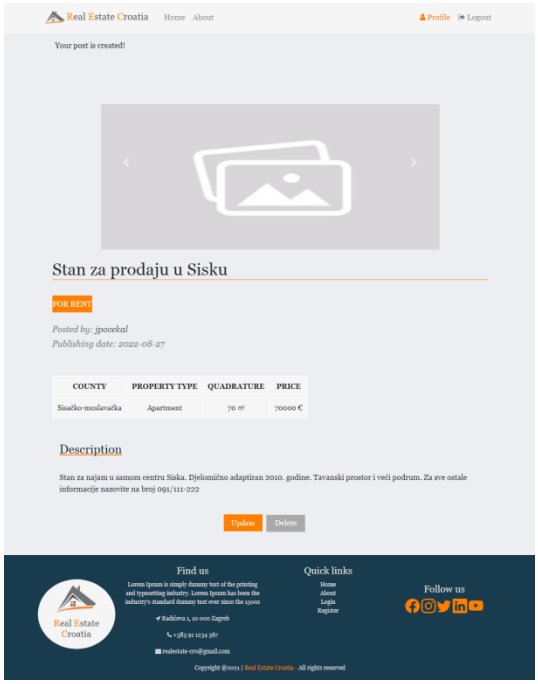
5.3.1. Kreiranje, ažuriranje i brisanje vlastitog oglasa

Kako bi korisnik kreirao oglas potrebno je u navigacijskoj traci odabrati *Profil* te u bočnoj traci izabrati *New Post*. Kada se otvori forma za kreiranje novog oglasa potrebno ju je popuniti ispravnim podacima i odabrati gumb *Post*.



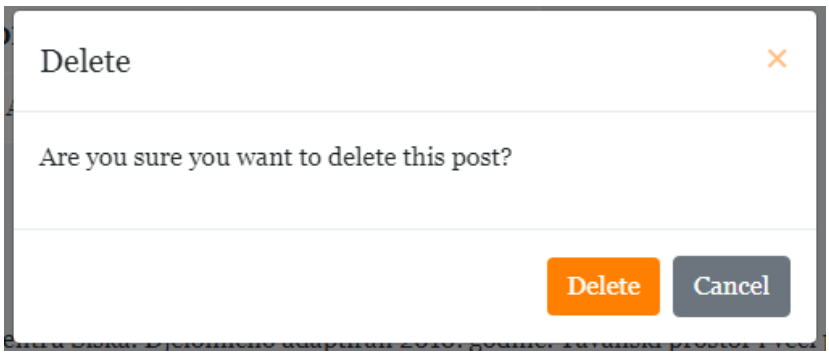
Slika 12 Kreiranje novog oglasa

Nakon što je forma uspješno ispunjena i oglas objavljen, aplikacija korisnika preusmjerava na stranicu koja prikazuje sve detalje novog oglasa.



Slika 13 Detalji oglasa

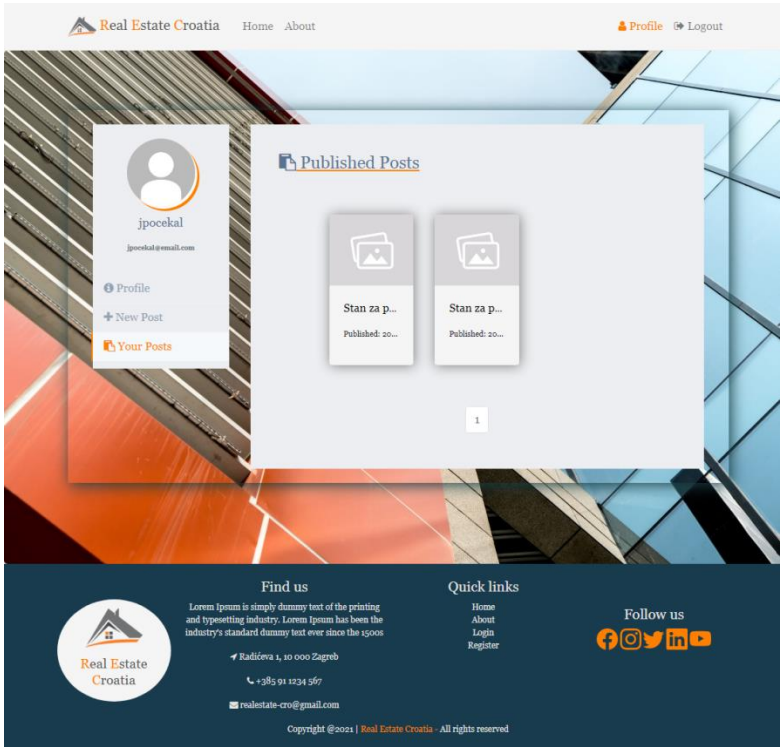
Ukoliko korisnik ne odabere slike prilikom kreiranje oglasa, biti će prikazana zadana (default) slika. Korisnik prilikom pregleda svojih oglasa ima mogućnost ažuriranja ili brisanja oglasa. Ukoliko odluči ažurirati informacije u oglasu, nakon pritiska na gumb *Update* korisnik će biti preusmjeren na formu za ažuriranje koja je već unaprijed popunjena postojećim podacima, a ukoliko odluči obrisati oglas, pritiskom na gumb *Delete* iskočit će prozor u kojem treba potvrditi da se oglas zaista želi obrisati.



Slika 14 Skočni prozor prilikom brisanja oglasa

5.3.2. Pregled oglasa

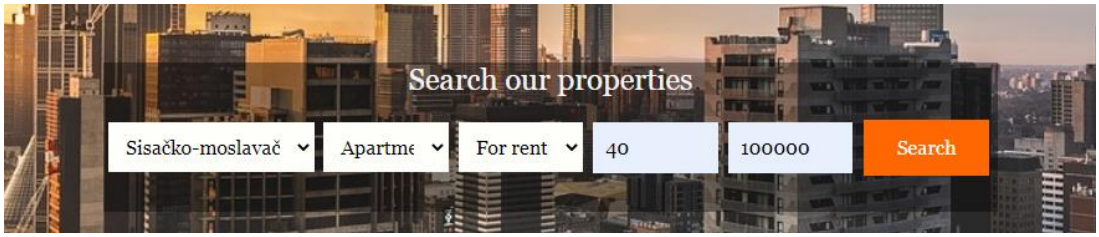
Korisnik ima mogućnost pregleda postojećih oglasa. Na početnoj stranici prikazani su svi oglasi na način da ih je po stranici prikazano 6. Stranice se mogu listati kako bi se pogledali ostali oglasi. Pritiskom na oglas prikazuju se informacije o njemu. Mogu se listati slike i vidjeti informacije o osobi koja je postavila oglas, no korisnik nema mogućnost ažuriranja ni brisanja tuđih oglasa. Ukoliko korisnik želi vidjeti sve svoje objavljene oglase može otići na svoj profil i u bočnoj traci izabrati *Your Posts*. Tamo će biti prikazani svi oglasi tog korisnika, a ukoliko želi vidjeti više informacija mora otvoriti oglas.



Slika 15 Pregled postojećih oglasa korisnika

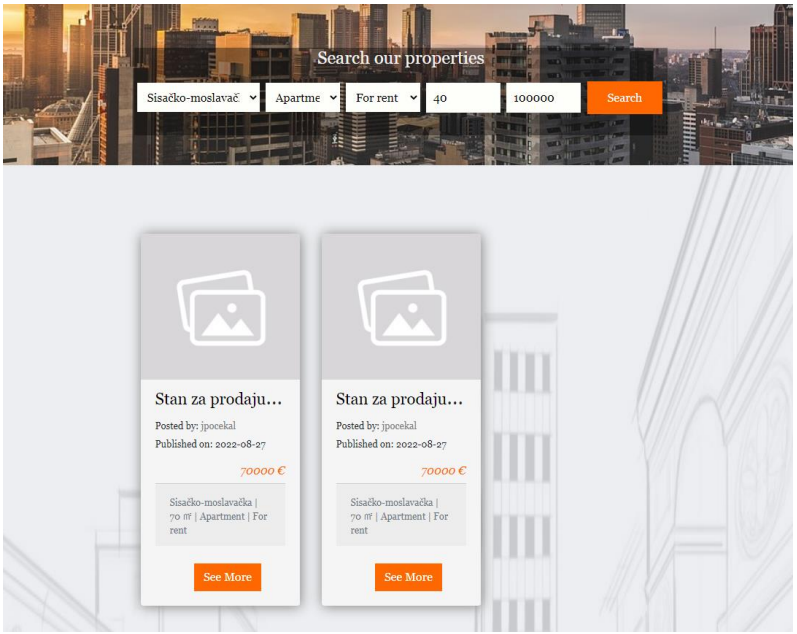
5.3.3. Filtriranje oglasa

S obzirom da na stranici može biti veliki broj oglasa bitna je značajka da se oglasi mogu filtrirati po određenim kriterijima kako bi korisnik što lakše i prije našao ono što ga interesira. Na početnoj stranici postoji mogućnost filtriranja oglasa prema županiji, po vrsti nekretnine (kuća/stan), po tipu (prodaja/najam), po minimalnoj kvadraturi i maksimalnoj cijeni (u eurima). Svi od navedenih kriterija se moraju ispuniti kako bi se prikazali oglasi koji će po svim kriterijima odgovarati potencijalnom kupcu.



Slika 16 Forma za filtriranje oglasa

Nakon pritiska na gumb *Search* prikazat će se svi oglasi koji odgovaraju unesenim kriterijima.



Slika 17 Filtrirani oglasi

5.4. Stranica s informacijama

Stranica *O nama* je dio gotovo svake web aplikacije. To je mjesto na kojem korisnik može saznati sve informacije o osnivanju, poslovnim uspjesima, vrijednostima, te kontakt informacije. Ukoliko korisnik Real Estate Croatia aplikacije želi saznati nešto od toga potrebno je da na navigacijskoj traci pritisne na *About* i aplikacije će ga preusmjerit na tu stranicu. Tamo može pronaći sve bitne informacije.

Slika 18 Stranica o nama

Također vrlo je bitno da informacije o kontaktu budu brzo pristupačne stoga je u podnožju svake stranice u aplikaciji postavljeno podnožje s informacijama i linkovima koji korisnika vode na sve društvene stranice i stranice aplikacije poput prijave i početne stranice.

6. Zaključak

Rezultat ovog završnog rada je sama aplikacija za prodaju i najam nekretnina u Hrvatskoj. Aplikacija je razvijena u Python programskom jeziku koristeći Flask framework. Baza podataka je implementirana u SQLite-u uz korištenje Flask-SQLAlchemy alata. Za stil je korišten CSS i Bootstrap, a za strukturu stranica se koristio HTML. Sav kod napisan je unutar PyCharm razvojnog okruženja.

S obzirom da je jedan od motiva razvoja ove aplikacije bio učenje Python programskog jezika, mogu potvrditi da je Python jezik koji se lako uči, sintaksa je vrlo jednostavna i kod je lako razumljiv. Prilikom izrade projekta pokušavala sam naći najbolje načine kako riješiti određeni problem pri čemu sam shvatila da je dokumentacija oko Python-a odlično napisana, postoji puno sadržaja na internetu, a zajednica programera pomaže pri svakom upitu.

Sama aplikacije je na početničkoj razini. Sadrži sve osnovne elemente jedne web aplikacije kao što su kreiranje, ažuriranje, brisanje i čitanje sadržaja, mogućnost registracije, prijave i odjave korisnika. Vizualno je modernija i responzivnog je dizajna. Smatram da su dijelovi poput filtriranja oglasa mogli biti bolje odrađeni u samom kodu koristeći Flask-SQLAlchemy funkcije. Aplikacija je lako proširiva i jednostavno se mogu dodati složenije mogućnosti.

Popis literature

1. JetBrains, *PyCharm*, Preuzeto 10.08.2022. s https://www.jetbrains.com/pycharm/
2. Nguyen, Q. (2019.), Hands-On Application Development with PyCharm: Accelerate your Python applications using practical coding tehniques in PyCharm, Birmingham, UK, Packt Publishing
3. Downey, A. B. (2014.), Naučite Python, Zagreb, Dobar Plan
4. Amazon Web Services (AWS), *What is Python*, Preuzeto 10.08.2022. s https://aws.amazon.com/what-is/python/
5. Grinberg, M. (2014.), Flask Web Development, USA, O'Reilly Media
6. Python Basics, *What is Flask Python*, Preuzeto 10.08.2022. s https://pythonbasics.org/what-is-flask-python/
7. The Pallets Projects, *Flask*, Preuzeto 10.08.2022. s https://palletsprojects.com/p/flask/
8. Haldar, S. (2007.), Inside SQLite, USA, O'Reilly Media
9. Codecademy, *What is SQLite*, Preuzeto 11.08.2022. s https://www.codecademy.com/article/what-is-sqlite
10. TutorialsPoint, *SQLite overview*, Preuzeto 13.08.2022. s https://www.tutorialspoint.com/sqlite/sqlite_overview.htm
11. SQLAlchemy, *SQLAlchemy 1.4. Documentation*, Preuzeto 13.08.2022. s https://docs.sqlalchemy.org/en/14/
12. Cvetković, D., Marković, D., Popović, R. (2010.), Multimedija, Sveučilište Singidunum, Beograd
13. Get Bootstrap, Preuzeto 15.08.2022. s https://getbootstrap.com/
14. MDN web docs, *HTML: HyperText Markup Language*, Preuzeto 16.08.2022. s https://developer.mozilla.org/en-US/docs/Web/HTML
15. PureVPN, *What is 127.0.0.1*, Preuzeto 19.08.2022. s https://www.purevpn.com/blog/what-is-127-0-0-1/
16. Real Python, *Use a Flask Blueprint to Architect Your Applications*, 24.08.2022. s https://realpython.com/flask-blueprint/

Popis slika