SVEUČILIŠTE U MOSTARU

**FAKULTET STROJARSTVA,RAČUNARSTVA**

**I ELEKTROTEHINKE**

**PROJEKTNA DOKUMENTACIJA**

**NEKRETNINE**

Dragan Drljepan

Frano Tomas

Slavko Ramljak

Jure Šarac

**Nekretnine**

**Sažetak:**

U projektnoj dokumentaciji je predstavljena izrada aplikacija za kupovinu nekretnina, opisan je odabir alata za rad i proces izrade aplikacije.

**Sadržaj**

[1. OPIS PROJEKTA 1](#_Toc148441629)

[2. LOGIN STRANICA 2](#_Toc148441630)

[3. ADMIN 4](#_Toc148441631)

[4. KORISNIK 5](#_Toc148441632)

[4.1. Početna stranica 5](#_Toc148441633)

[4.2. Uređivanje profila 7](#_Toc148441634)

[5. AGENCIJA 11](#_Toc148441635)

[5.1. Prikaz nekretnina 11](#_Toc148441636)

[5.2. Uređivanje nekretnina 12](#_Toc148441637)

[5.3. Dodavanje nove nekretnine 13](#_Toc148441638)

[6. SOCKET.IO 14](#_Toc148441639)

# OPIS PROJEKTA

Aplikacija za nekretnine nam omogućava prijavu ili registraciju novog korisnika. Imamo 3 tipa korisnika a to su : Admin, Agencija i Korisnik. Admin ima pregled svih registriranih korisnika, Agencija ima mogućnost pregleda svih nekretnina, dodavanje i uređivanje nekretnina i pregled upita odnosno razgovora za nekretnine a Korisnik ima mogućnost pregleda ponuđenih nekretnina za kupnju gdje može ući u nekretninu pogledati detaljniji opis o nekretnina i pokrenuti razgovor sa agencijom koja prodaju određenu nekretninu i korisnik ima mogućnost uređivanja svog profila.

Za izradu aplikacije smo koristili :

* PHP
* Socket.io
* CSS i Bootstrap
* MySQL
* NodeJS server

# LOGIN STRANICA

Kod za login stranicu se nalazi u *indeks.php* fileu, predstavlja sustav prijave za web stranicu koji koristi PHP i MySQL. Glavna svrha sustava prijave je omogućiti korisnicima da se prijave u svoje korisničke račune i pristupe svojim osobnim informacijama. Koristi se sesija kako bi se održavalo stanje prijave korisnika tijekom njihove sesije na web stranici.

**Funkcionalnosti**

1. Sustav sesija: Provjera stanja prijave korisnika pomoću PHP sesija kako bi se osiguralo da samo prijavljeni korisnici mogu pristupiti određenim dijelovima web stranice.

2. Forma za prijavu: Omogućava korisnicima unos korisnika i lozinke, kao i odabir uloge pri prijavi.

3. Poruke upozorenja: Prikazuje upozorenje ako postoji greška pri prijavi ili neuspješnom pokušaju prijave.

4. Povezivanje s bazom podataka: Pretpostavlja se da postoji baza podataka koja pohranjuje korisničke informacije, ali sama veza s bazom podataka nije uključena u ovom kodu.

5. Preusmjeravanje korisnika: Ako je korisnik već prijavljen, sustav će automatski preusmjeriti korisnika na određenu početnu stranicu ovisno o ulozi korisnika.

**Uputstva za upotrebu**

1. Osigurajte da je PHP postavljen i konfiguriran na vašem web poslužitelju.

2. Stvorite odgovarajuću bazu podataka i tablice za pohranu korisničkih podataka.

3. Prilagodite kod tako da odgovara strukturi vaše baze podataka i sigurnosnim zahtjevima vašeg projekta.

4. Osigurajte da su svi korisnički unosi provjereni i sigurni kako biste spriječili ranjivosti poput SQL injekcija ili XSS napada.

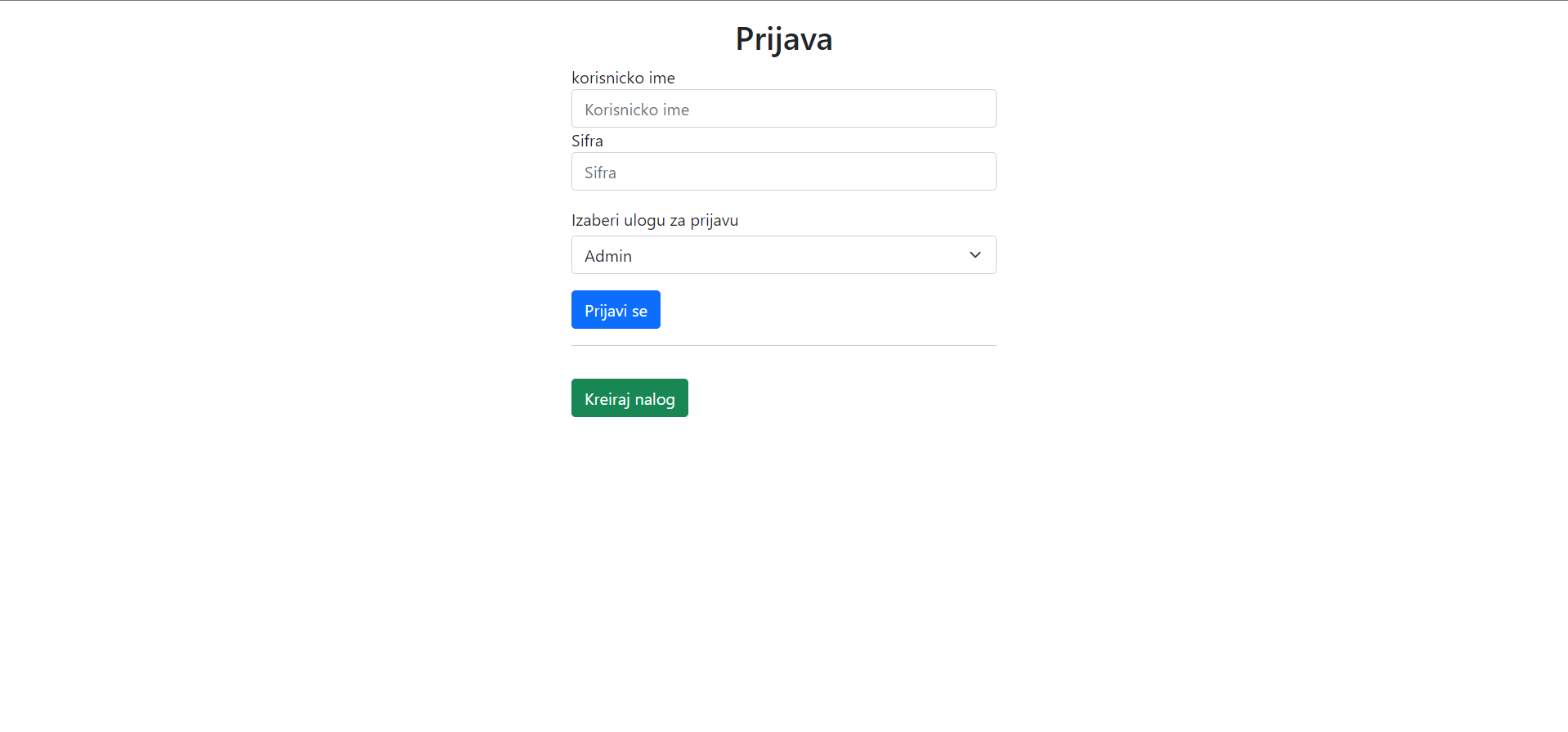
5. Prilagodite stilove i sučelje kako biste odgovarali vašim specifičnim potrebama dizajna.

6. Instalirajte odgovarajuću verziju Node JS (21.2.0).

7. Unutar foldera 'app' pokrenite php server sa naredbom 'php -S localhost:8000'

7. Unutar foldera 'nekretnine-chat-server' pokrenite naredbu 'npm install' kako bi ste instalirali potrebne pakete za pokretanje servera.

8. Nakon toga unutar istog foldera možete pokrenuti server sa naredbom 'npm run devStart '.



Login stranica

# ADMIN

Ovaj kod predstavlja administratorsku kontrolnu ploču za upravljanje korisničkim računima u sustavu. Administratorska kontrolna ploča omogućuje administratorima pregled, uređivanje i brisanje korisničkih podataka pohranjenih u MySQL bazi podataka. Koristi se PHP sesija za provjeru autentifikacije, a Bootstrap se koristi za poboljšanje korisničkog sučelja.

**Funkcionalnosti**

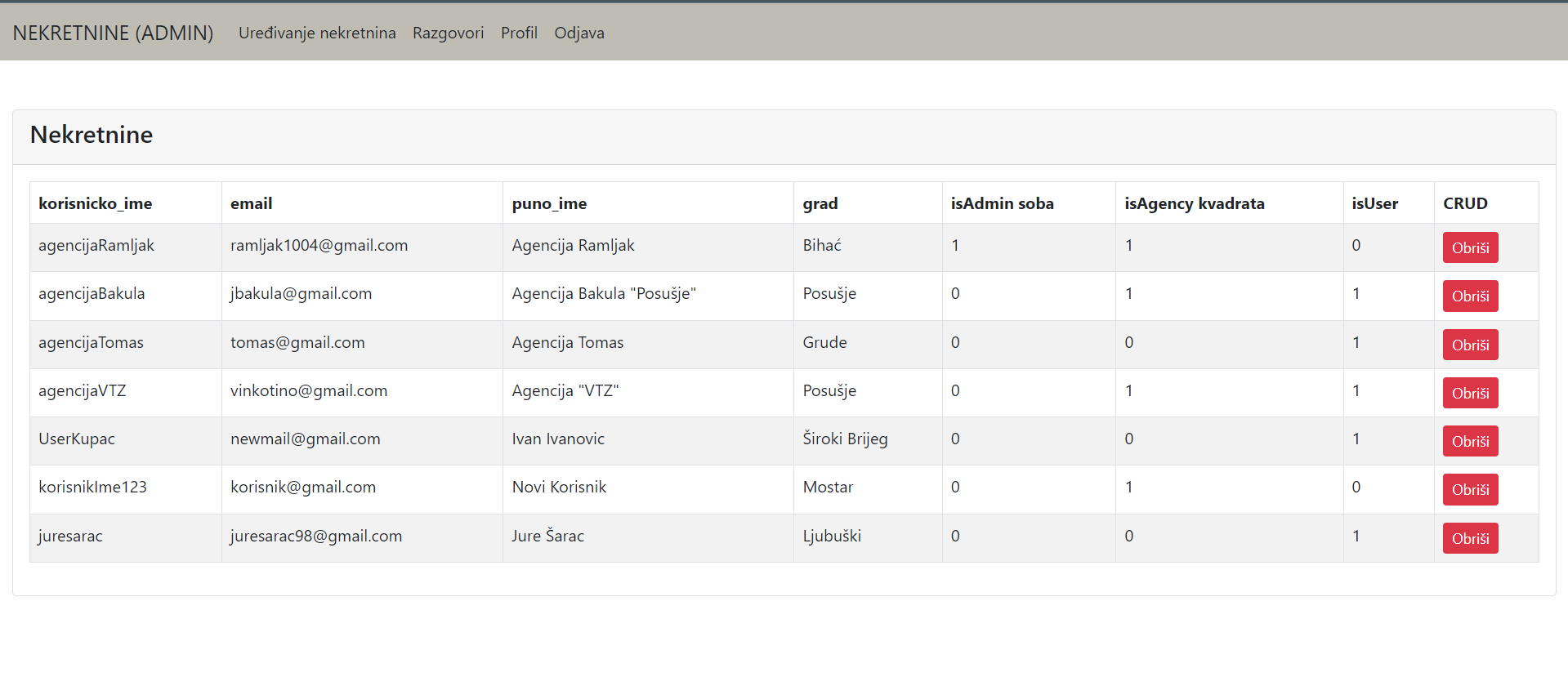
1. Sustav prijave i autentifikacije: Kod počinje provjerom sesija kako bi se osiguralo da samo prijavljeni administratori mogu pristupiti ovoj kontrolnoj ploči.

2. Navigacijska traka: Navigacijska traka omogućuje administratorima jednostavan pristup različitim dijelovima administratorske kontrolne ploče, uključujući upravljanje nekretninama, razgovorima i profilom.

3. Prikaz korisnika: Podaci svih korisnika prikazani su u obliku tablice, omogućujući administratorima brz pregled ključnih informacija o svakom korisniku, kao što su korisničko ime, e-mail adresa, puno ime, grad te njihove uloge.

4. Upravljanje korisnicima: U okviru svakog reda tablice, administratorima se omogućuje brisanje određenog korisničkog računa putem gumba za brisanje.

Ovaj kod pruža temeljnu administratorsku kontrolnu ploču za upravljanje korisničkim računima u sustavu. Implementirane funkcionalnosti omogućuju administratorima učinkovito upravljanje korisnicima, pregled njihovih podataka i brisanje računa po potrebi. Daljnji razvoj sustava mogao bi uključivati dodatne značajke poput uređivanja korisničkih podataka ili dodavanja novih korisničkih privilegija.



Prikaz stranice za prijavljenog admina

# KORISNIK

## Početna stranica

1. Inicijalizacija sesije i uključivanje datoteka:

-`session\_start()`: Inicijalizira novu sesiju ili nastavlja postojeću sesiju.

- `include "../php/db\_conn.php";`: Uključuje datoteku `db\_conn.php`, koja vjerojatno sadrži logiku povezivanja s bazom podataka.

2. HTML struktura:

- osnovna struktura HTML-a koja uključuje navigacijsku traku i glavni sadržaj stranice.

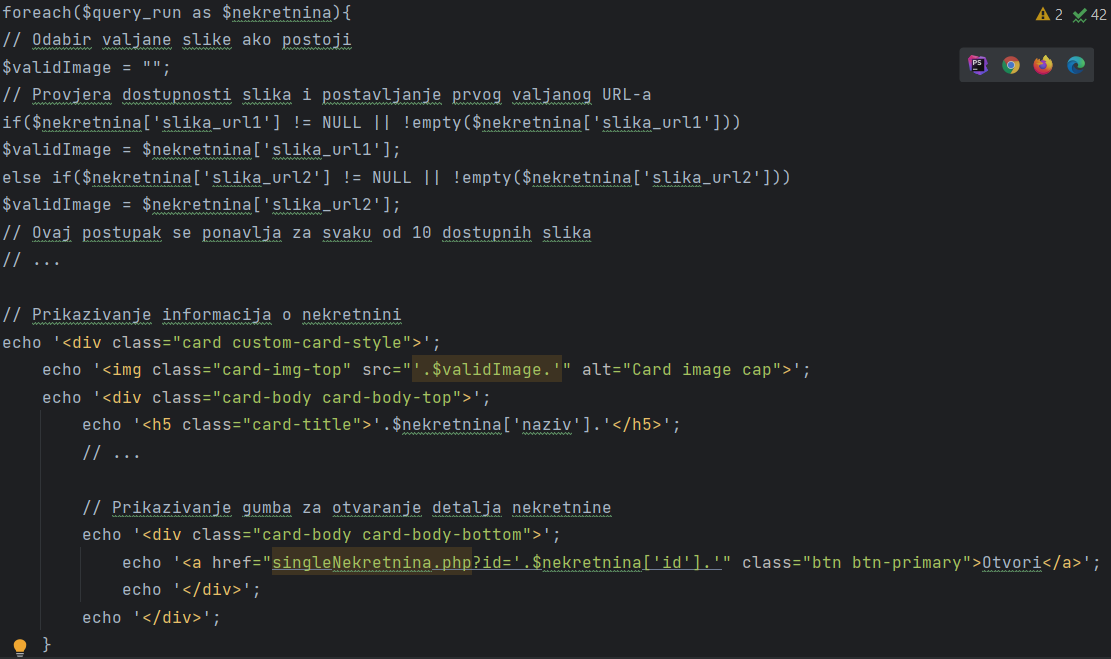
3. Navigacija

- prikazuje navigacijsku traku s opcijama "Razgovori", "Profil" i "Odjava" kako bi se omogućila navigacija kroz različite dijelove aplikacije.

4. Forma za filtriranje nekretnina

- Omogućuje korisnicima unos različitih kriterija za filtriranje nekretnina, uključujući naziv, cijenu, lokaciju, broj soba, broj kvadrata, stanje itd.

5. Petlja za prikazivanje rezultata pretrage:



6. PHP logika za filtriranje nekretnina

- procesira podatke poslane putem POST zahtjeva iz forme za filtriranje te generira SQL upit temeljen na unesenim filterima.

7. Prikaz rezultata pretrage

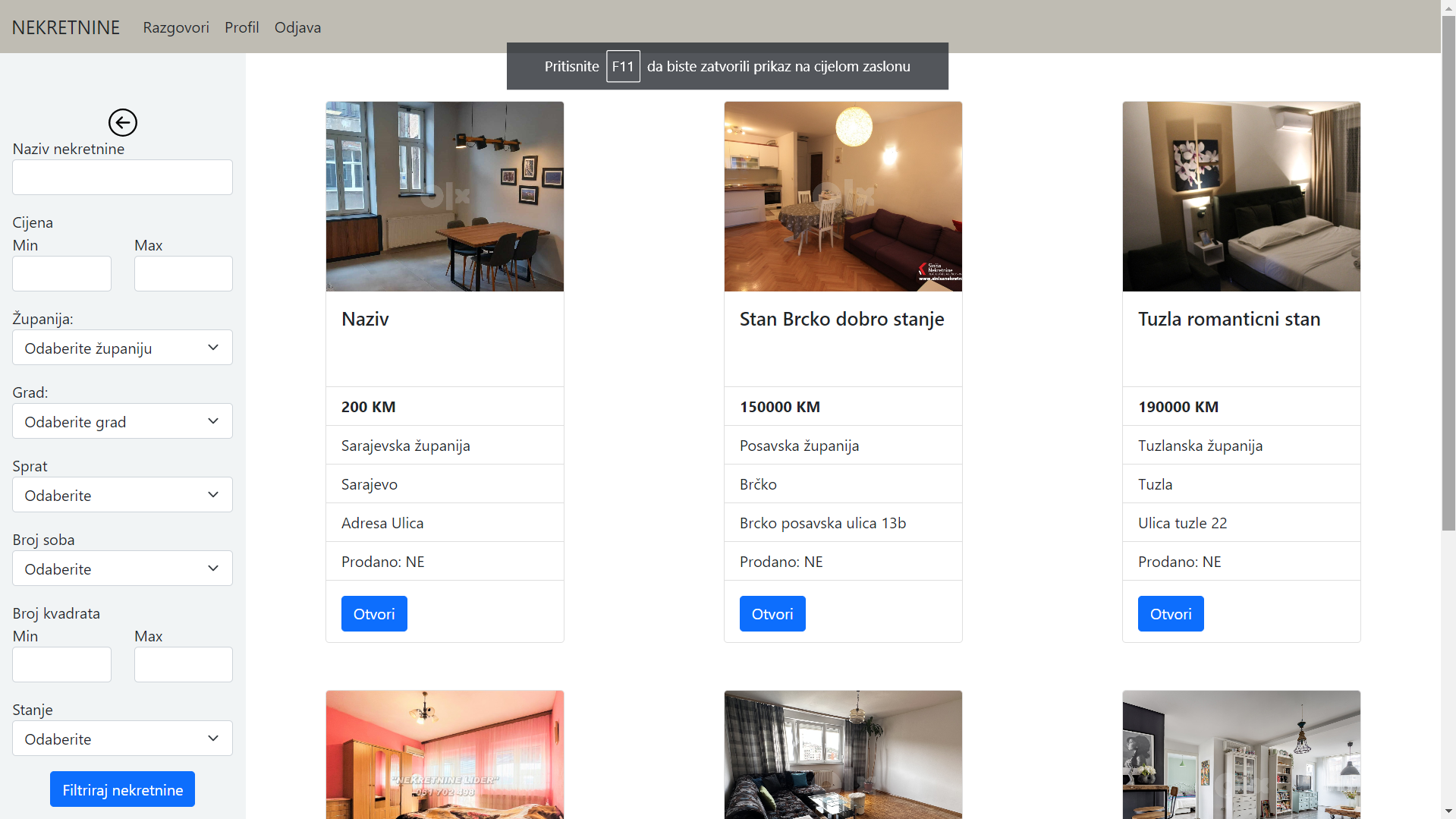
- izvršava se SQL upit s primijenjenim filtrima te rezultati prikazuju u obliku kartica koje sadrže relevantne informacije o svakoj nekretnini, uključujući naziv, cijenu, lokaciju, opis, broj soba, stanje i slike.

8. Upravljanje sesijom

- ako sesija nije valjana, korisnik se preusmjerava na prethodno valjanu URL adresu ili početnu stranicu kako bi osigurao sigurnost i pouzdanost sesije.

9. Dinamički upiti za prikaz slika

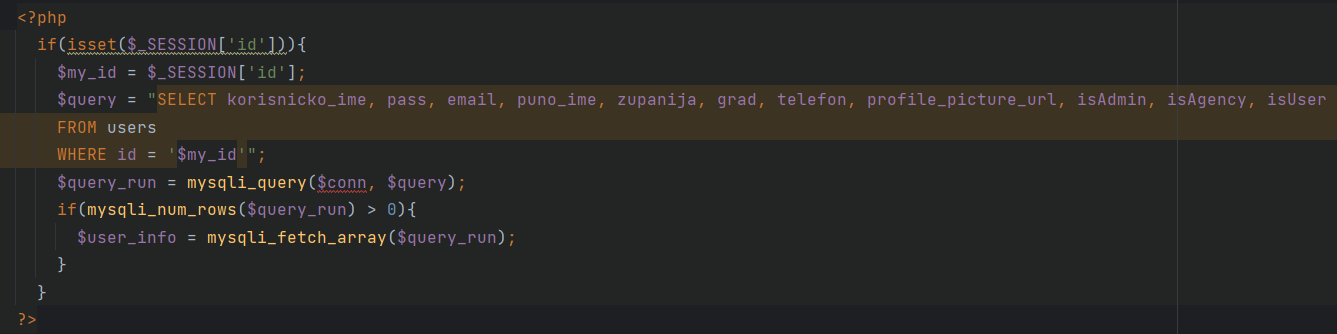
- provjerava dostupnost slika nekretnina putem URL-ova te prikazuje prvu dostupnu sliku za svaku nekretninu kako bi se poboljšao vizualni dojam i privukla pozornost korisnika.



Prikaz početne stranice prijavljenog korisnika

## Uređivanje profila

1. Prikaz informacija o korisniku u formi



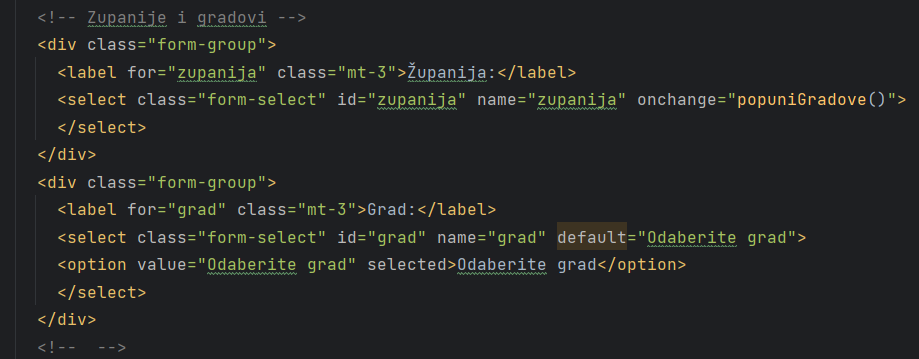
Ovaj dio koda prvo provjerava postoji li sesija. Ako sesija postoji, dohvaćaju se informacije o korisniku iz baze podataka na temelju ID-ja pohranjenog u sesiji. Ove informacije uključuju korisničko ime, šifru, e-poštu, puno ime, županiju, grad, telefon, URL slike profila te informacije o korisnikovim ovlastima (isAdmin, isAgency, isUser).

2. Forma za uređivanje profila



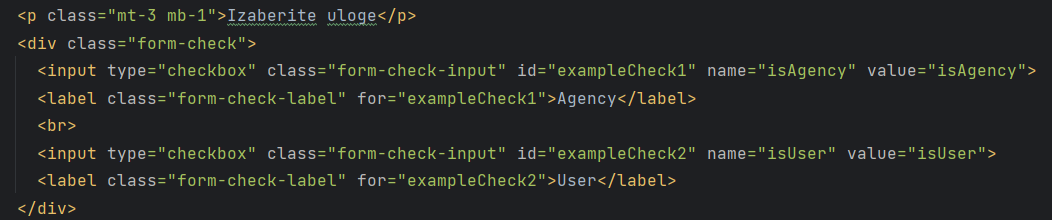
Ovdje je prikazana forma za uređivanje profila. Omogućuje korisniku promjenu određenih osobnih podataka, poput imena, telefonskog broja, odabira uloga (npr. Agencija, Korisnik) te omogućuje korisniku da odabere novu sliku profila.

3. Dinamičko punjenje županija i gradova



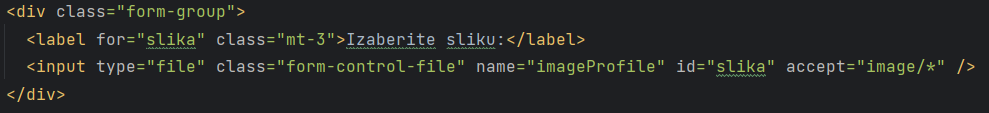
Ove dva dijela koda omogućuju korisniku odabir županije i grada. Kada se odabere županija, poziva se JavaScript funkcija `popuniGradove()` koja dinamički puni popis gradova ovisno o odabranoj županiji. Ovo poboljšava korisničko iskustvo i olakšava odabir točne lokacije.

4. Odabir uloga



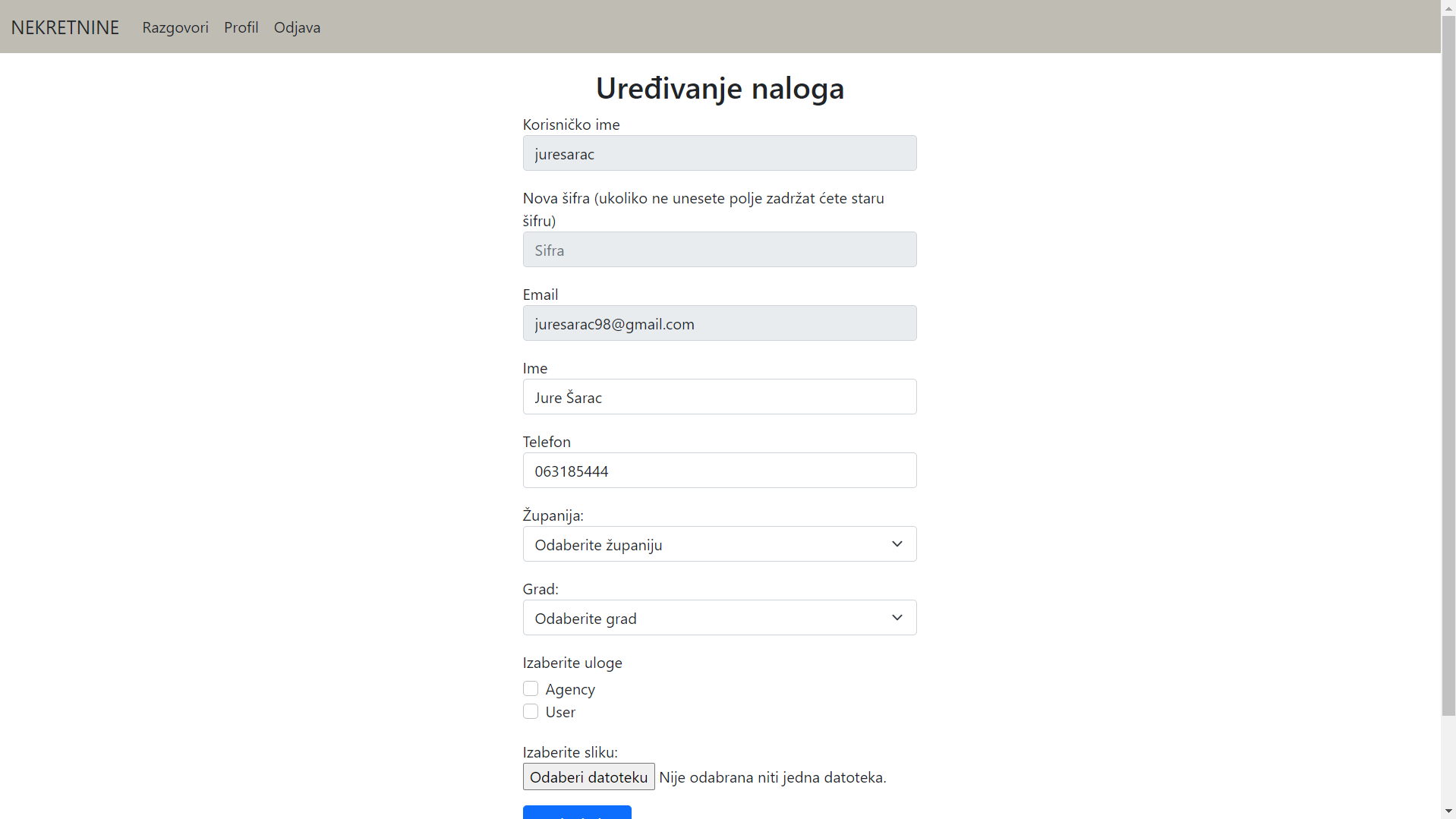
Ovaj dio koda omogućuje korisniku odabir uloge. Korisnik može odabrati više uloga pritiskom na odgovarajuće kućice za potvrdu. U ovom slučaju, korisnik može odabrati između "Agency" i "User". Ovaj odabir uloga omogućava aplikaciji da prilagodi funkcionalnosti korisniku na temelju odabrane uloge.

5. Dodavanje slike profila



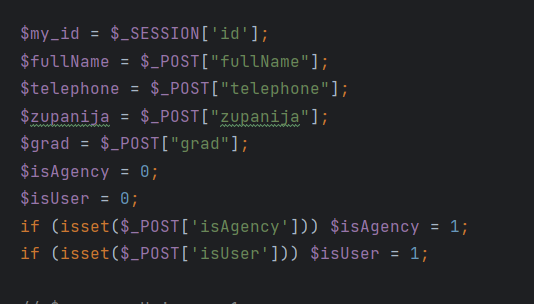
Ovaj dio koda predstavlja polje za unos slike profila. Korisnik može odabrati sliku s lokalnog uređaja pritiskom na gumb za pregledavanje. Atribut `accept="image/\*"` osigurava da korisnik može odabrati samo slikovne datoteke prilikom dodavanja slike profila.

Ova objašnjenja pomažu u razumijevanju dijelova koda koji se odnose na prikaz informacija o korisniku te na formu za uređivanje profila, uključujući dinamičko punjenje županija i gradova, dodavanje uloga i slike profila.



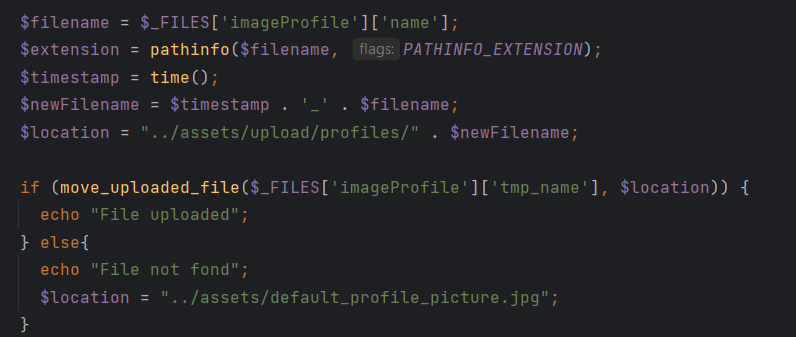
*profileCode.php* sadrži skriptu koja se izvršava nakon podnošenja obrasca za uređivanje naloga.

-Postavljanje varijabli na osnovu podataka obrasca:

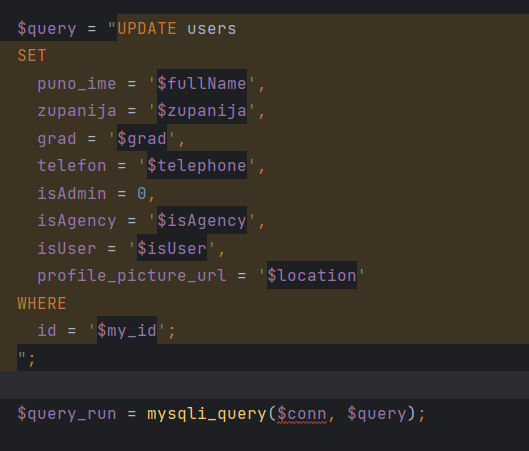


-Provjera popunjenosti obaveznih polja i preusmjeravanje korisnika na odgovarajuće stranice u slučaju praznih polja ili drugih pogrešaka.

-Učitavanje i spremanje slike profila na server:



-Ažuriranje korisničkog profila u bazi podataka na temelju podataka obrasca:



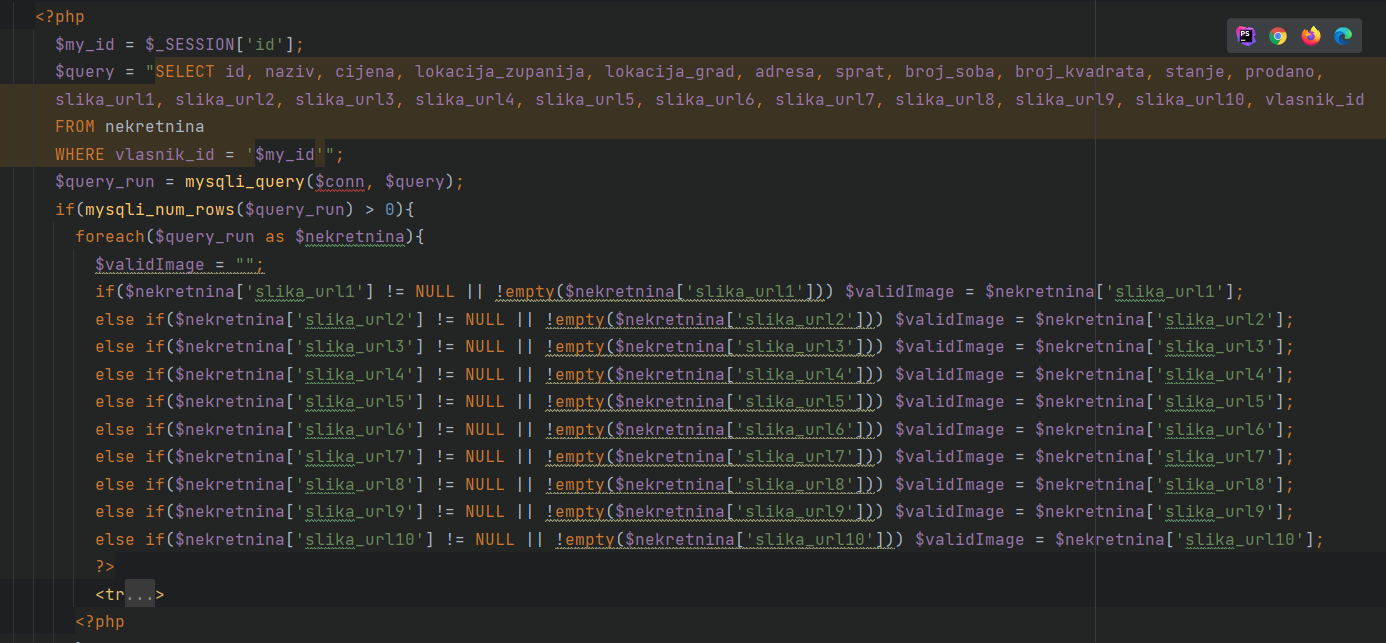
-Provjera izvršavanja upita i preusmjeravanje korisnika na odgovarajuću stranicu s porukom o uspješnom ili neuspješnom ažuriranju profila.

# AGENCIJA

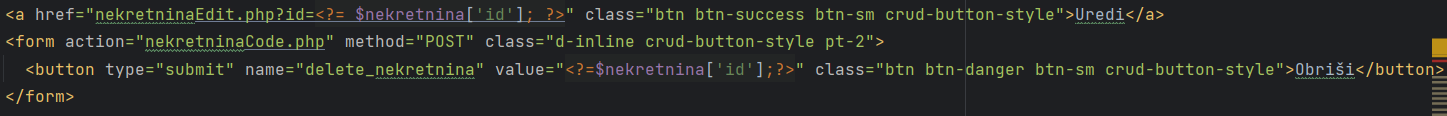
Kada se korisnik prijavi ako agencija na početnoj stranici ima pregled svih nekretnina kao obični korisnik ali agencija ima pristup uređivanju i dodavanju novih nekretnina.

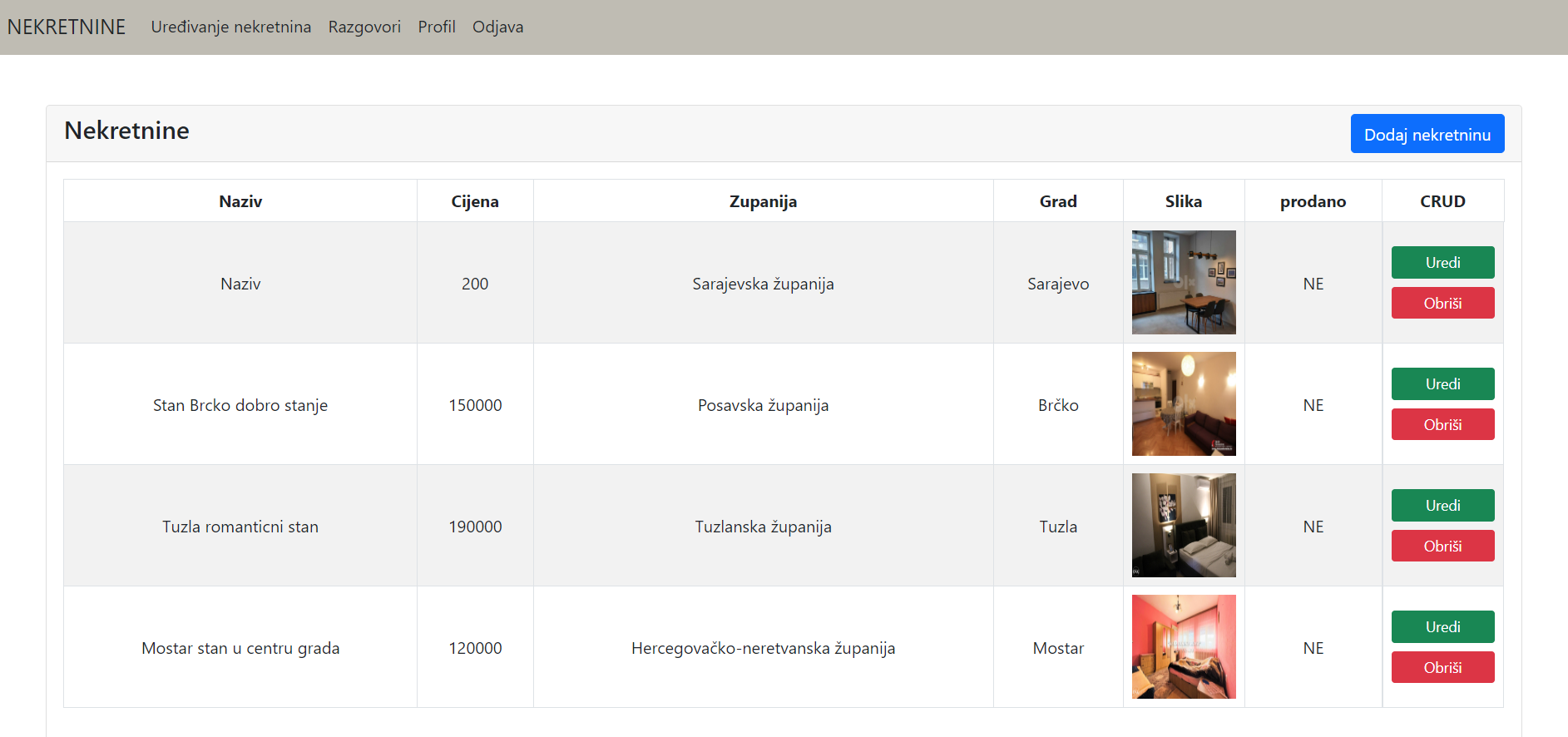
## Prikaz nekretnina

- Dohvaćanje nekretnina koje pripadaju određenoj agenciji iz baze podataka i njihovo prikazivanje u tablici:



-Omogućavanje uređivanja ili brisanja pojedinačnih unosa nekretnina povezanih s agencijom:





## Uređivanje nekretnina

Za uređivanje nekretnina smo koristili formu :

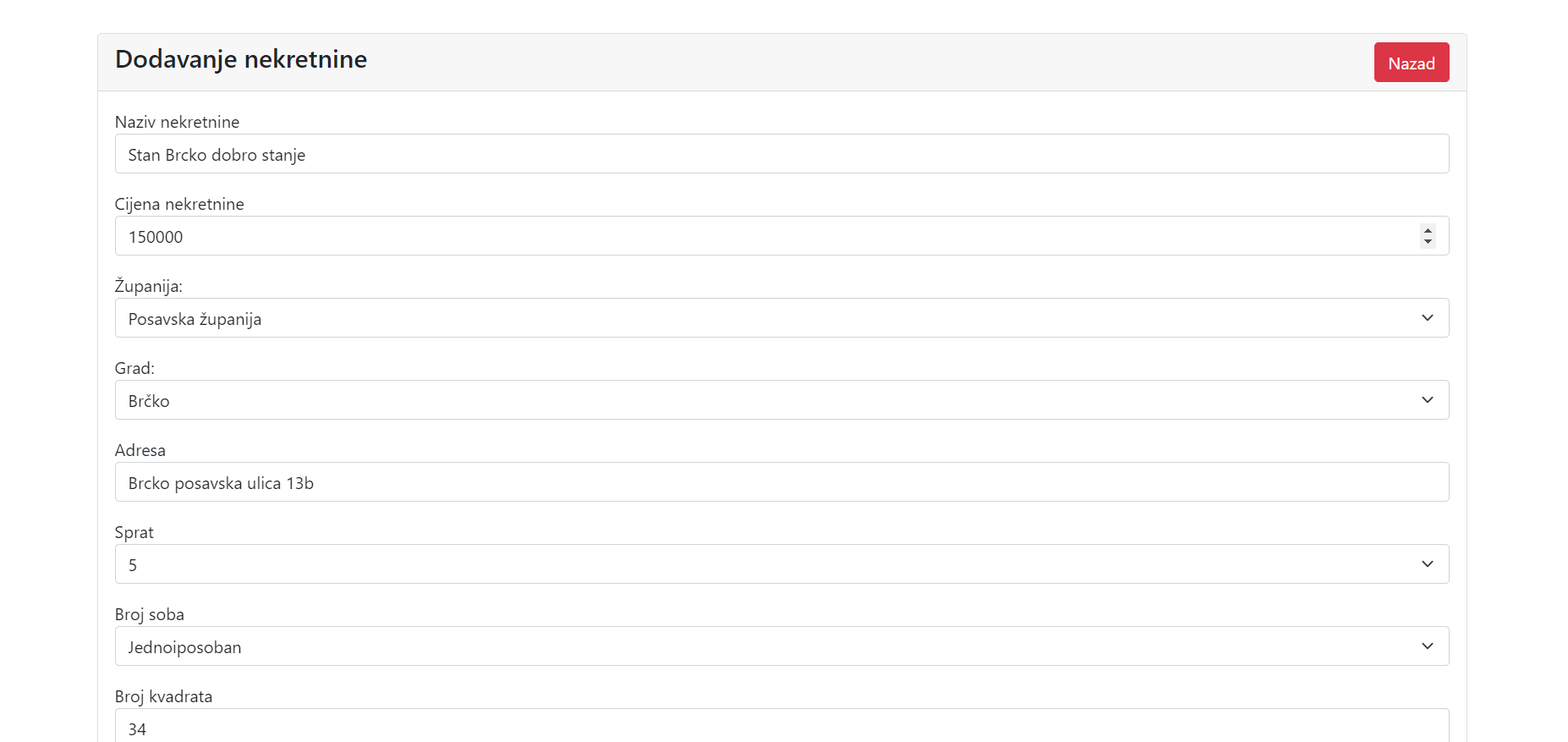
<form action="nekretninaCode.php" method="POST" enctype="multipart/form-data">: Ova linija pokreće skriptu "nekretninaCode.php" kada se forma pošalje, koristeći metodu POST za slanje podataka. Također, "enctype" postavljen je na "multipart/form-data" kako bi se podržao unos datoteka.

<div class="form-group">: Ovaj element definira grupu elemenata forme kako bi se pravilno stilizirali.

<label>Naziv nekretnine</label>: Ovo je oznaka koja označava polje unosa. U ovom slučaju, označava se unos za naziv nekretnine.

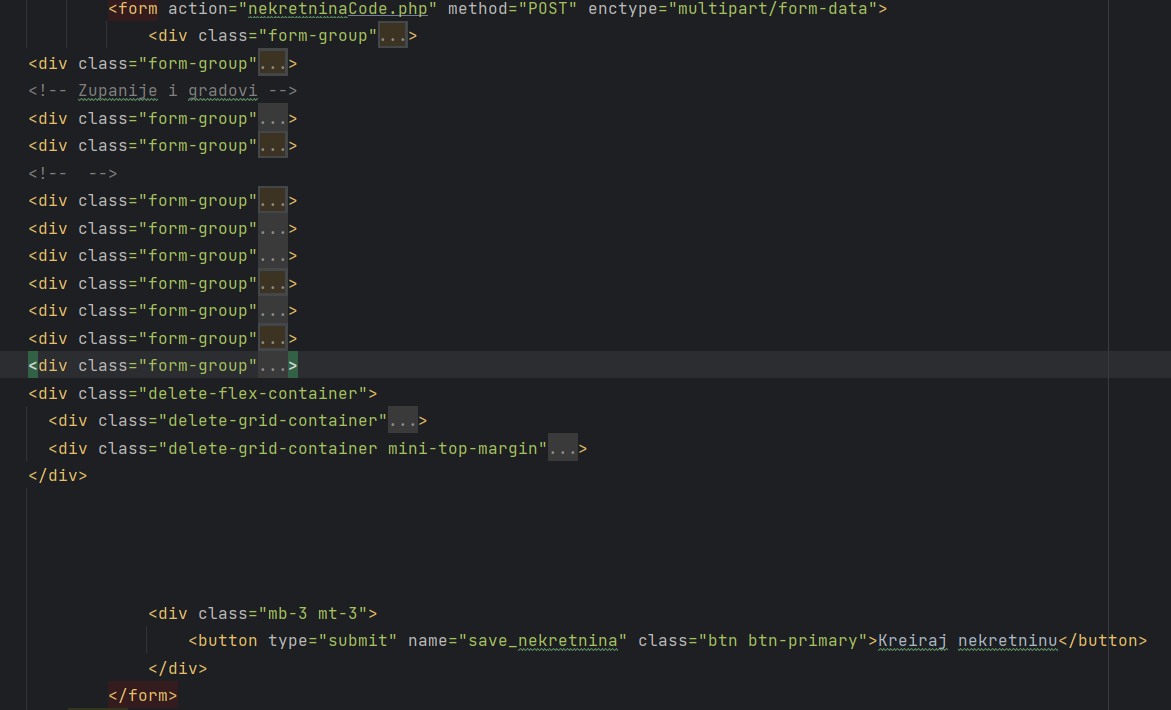
<input type="text" name="naziv" class="form-control" value="<?=$nekretnina['naziv'];?>">: Ovo polje za unos omogućuje korisniku da unese naziv nekretnine. "name" atribut je ključ koji se koristi za identifikaciju ovog polja u skripti za obradu obrasca. "class" atribut je postavljen na "form-control" za stiliziranje, a "value" atribut je postavljen na trenutnu vrijednost naziva nekretnine iz baze podataka kako bi se omogućilo uređivanje.

<button type="submit" name="update\_nekretnina" class="btn btn-primary">Uredi nekretninu</button>: Ovaj gumb omogućuje korisniku da pošalje formu. "name" atribut je postavljen na "update\_nekretnina" kako bi se odredila akcija koja se pokreće kada se forma pošalje.



## Dodavanje nove nekretnine

Dio koda koji se odnosi na formu za kreiranje nove nekretnine:



U ovom kodu:

- `action="nekretninaCode.php"` označava datoteku na koju će se poslati podaci nakon slanja obrasca.

- `method="POST"` definira HTTP metodu za slanje podataka obrasca.

- `enctype="multipart/form-data"` omogućuje slanje datoteka putem obrasca.

- `div` elementi s klasom "form-group" služe za oblikovanje elemenata obrasca.

- `input` elementi pružaju polja za unos različitih podataka o nekretnini.

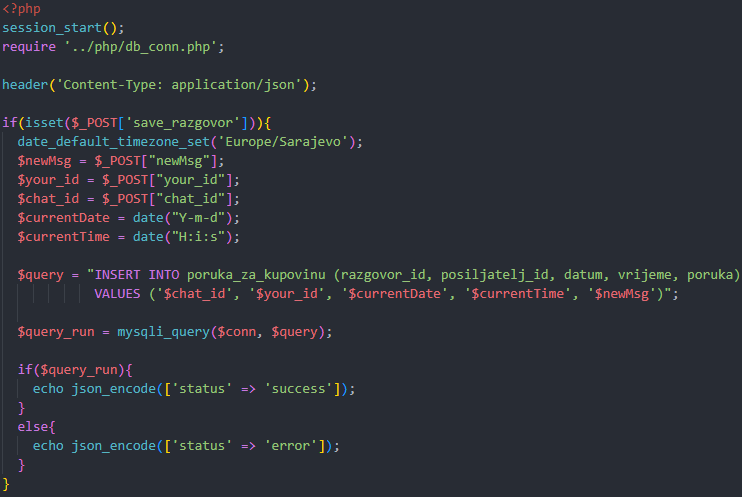
- `button` element s klasom "btn btn-primary" predstavlja gumb za slanje obrasca.

Ovaj kod omogućuje korisnicima da unesu podatke o nekretnini i pošalju ih putem obrasca na `nekretninaCode.php` za daljnju obradu.

# WEB SOCKET I CHAT

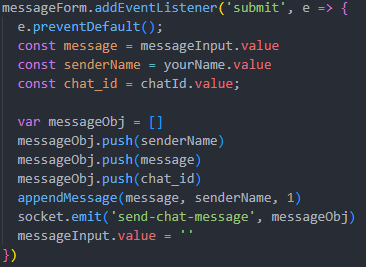
Slanje poruke unutar razgovora sastoji se od dvije komponente.

Prva komponenta je slanje poruke u bazu podataka. Što je potrebno za spremanje poruke kako bi se u budućnosti mogla ponovno pogledati. U bazu se spremaju svi potrebni podaci o poruci: id razgovora, posiljatelj, datum, vrijeme, sadržaj poruke.

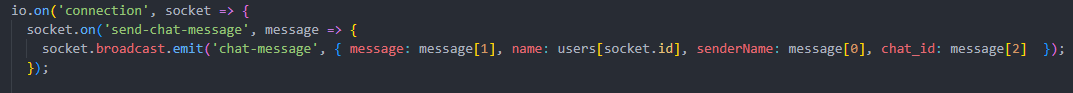


Samo spremanje poruke u bazu neće omogućiti trenutni prikaz poruke na ekranima korisnika koji su uključeni u razgovoru. Da bi vidjeli poruke morali bi osvježiti prozore.

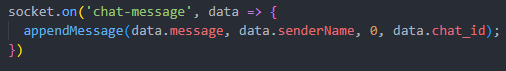
Za potrebe prikazivanja poruke na ekranima odmah nakon slanja poruke potrebno je koristiti web socket. To je druga komponenta poruka osim što se sprema u bazu također se putem web scoketa šalje na Node server.



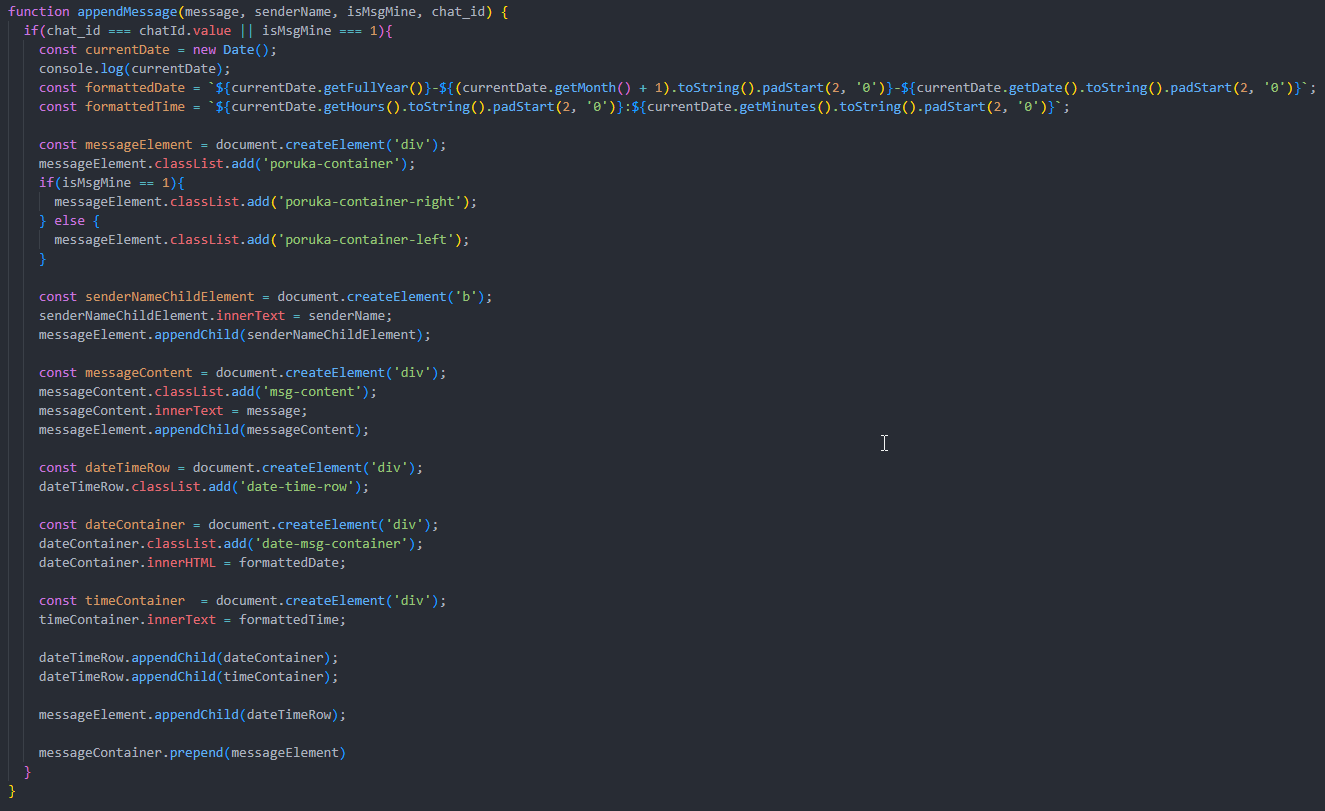
Ovo je klijentski dio koda koji šalje poruke Node serveru na ključnu riječ 'send-chat-message'.



Server cijelo vrijeme „sluša“ i kada god primi poruku on tu istu poruku šalje klijentima kod kojih je tu poruku potrebno ispisati istu na ekran.



Klijent websocket poruke od servera prima u ovom dijelu koda unutar kojega pokreće funkciju 'appendMessage' koja će obaviti zadatak ispisivanja poruke na ekran.



Ovako izgleda funkcija za ispis poruke, ukratko objašnjene izgleda ovako:

* Uzima se trenutni datum i vrijeme
* Kreiraju se određeni <div>, <b> elementi i na njih se dodavaju odgovarajuće klase koje imaju definiran izgled unutar .css datoteke.
* Klase 'poruka-container-right' i 'poruka-container-left' se dodavaju u zavisnosti od toga da li je poruka naša ili od drugog korisnika.
* Kreiranu poruku u obliku <div> … </div> dodavamo na glavni <div> element koji sadrži sve ostale poruke.

const messageContainer = document.getElementById(**'razgovorIdDiv'**)