Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

ReactJS

Domaći zadatak 3

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
|  |  |  |
| Link ka Github-u |  | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc117932679)

[2 Implementacija 3](#_Toc117932680)

[3 Korisničko uputstvo 3](#_Toc117932681)

# Korisnički zahtev

Potrebno je kreirati aplikaciju koja ce da olakša korisnicima pretragu, pregled i kontaktiranje agencije oko dostupnih nekretnina. Aplikacija treba da omogući korisnicima da pretražuju nekretnine prema različitim kriterijumima, kao što su tip nekretnine (stan, kuća, apartman, vikendica, poslovni prostor), cena, broj spavaćih soba, opis i slično. Takođe, korisnici treba da mogu da filtriraju rezultate pretrage po tipu nekretnine i ceni. Za svaku nekretninu, treba prikazati osnovne informacije kao što su naslov, opis, cena, broj spavaćih soba i tip nekretnine.

Pored toga, korisnici treba da mogu da kontaktiraju agenciju ili prodavca nekretnina putem obrasca za kontakt. Poruke koje korisnici pošalju treba da budu sačuvane i prikazane u tabelarnom formatu kako bi ih agencija ili prodavac mogli pregledati. Aplikacija takođe treba da omogući brisanje poruka iz tabele.

# Implementacija

Komponenta "HomePage" predstavlja početnu stranicu naše aplikacije za upravljanje nekretninama. Ova stranica ima za cilj da dočara korisnicima suštinske informacije o našoj agenciji za nekretnine.

import React, { useState, useEffect } from 'react';

import './HomePage.css';

import Navbar from '../Navbar/Navbar';

import Footer from '../Footer/Footer';

const HomePage = () => {

  const [randomImages, setRandomImages] = useState([]);

  const [isLoading, setIsLoading] = useState(true);

  // const fetchRandomImages = async () => {

  //   const accessKey = '\_dcS1nPF6IIONuQrJCArcuCg6u2EpN876eg2wHKZTNQ';

  //   const keywords = ['house', 'apartment', 'architecture', 'interior', 'building'];

  //   const numberOfImages = 10;

  //   const randomKeyword = keywords[Math.floor(Math.random() \* keywords.length)];

  //   try {

  //     const response = await fetch(

  //       `https://api.unsplash.com/photos/random?count=${numberOfImages}&query=${randomKeyword}`,

  //       {

  //         headers: {

  //           Authorization: `Client-ID ${accessKey}`,

  //         },

  //       }

  //     );

  //     if (response.ok) {

  //       const data = await response.json();

  //       setRandomImages(data);

  //       setIsLoading(false);

  //     } else {

  //       console.error('Failed to fetch random images');

  //       setIsLoading(false);

  //     }

  //   } catch (error) {

  //     console.error('Error:', error);

  //     setIsLoading(false);

  //   }

  // };

  // useEffect(() => {

  //   fetchRandomImages();

  // }, []);

  // if (isLoading) {

  //   return <div>Loading...</div>;

  // }

  return (

    <>

    <Navbar></Navbar>

    <div className="about">

      <header className="about-header">

        <h1>About Our Real Estate Agency</h1>

        <p>Trusted by Thousands Since 1999</p>

      </header>

      <section className="about-intro">

        <h2>Welcome to Our Agency</h2>

        <p>Our mission is to empower clients to make smart decisions about their home purchase or sale.</p>

      </section>

      <section className="about-content">

        <h3>Who We Are</h3>

        <p>We are a team of dedicated real estate professionals with over two decades of experience in helping clients find their dream homes.</p>

        <h3>What We Offer</h3>

        <p>Our portfolio includes a wide range of properties to fit every lifestyle and budget, from cozy apartments to luxury estates.</p>

        <h3>Why Choose Us</h3>

        <p>Our commitment to excellence and our client-first approach ensure that your buying or selling experience is seamless and rewarding.</p>

      </section>

      <section className="about-values">

        <h2>Our Values</h2>

        <div className="values-list">

          <div className="value">

            <h4>Integrity</h4>

            <p>We believe in honest and transparent dealings with all our clients.</p>

          </div>

          <div className="value">

            <h4>Expertise</h4>

            <p>Our agents are highly skilled and knowledgeable about the real estate market.</p>

          </div>

          <div className="value">

            <h4>Commitment</h4>

            <p>Your satisfaction is our top priority, and we work tirelessly to meet your needs.</p>

          </div>

        </div>

      </section>

      <section className="image-gallery">

        <h2>Random Real Estate Images</h2>

        {/\* <div className="gallery">

          {randomImages.map((image) => (

            <img

              key={image.id}

              src={image.urls.regular}

              alt={image.alt\_description}

              className="gallery-image"

            />

          ))}

        </div> \*/}

      </section>

    </div>

    <Footer></Footer>

    </>

  );

};

export default HomePage;

Komponenta "ContactForm" omogućava korisnicima da kontaktiraju našu agenciju za nekretnine putem jednostavnog obrasca za slanje poruke. Ova komponenta pruža korisnicima mogućnost da unesu svoje ime, email adresu i poruku kako bi postavili pitanja, izrazili interesovanje ili zatražili dodatne informacije o nekretninama.

Kada korisnik popuni formu i klikne na dugme "Submit", podaci se validiraju, i ako su svi potrebni podaci ispravno uneti, poruka se šalje. Sve poslate poruke čuvaju se u obliku niza podataka, omogućavajući našoj agenciji da pregleda i odgovori na svaku pojedinačnu poruku. Takođe, korisniku se prikazuje odgovarajuća obaveštenja o statusu slanja poruke, uključujući obaveštenje o uspešnom slanju ili zahtev za popunjavanjem svih obavezanih polja.

import React, { useState } from 'react';

import './ContactForm.css';

import TextInput from './TextInput'; // Importujemo TextInput komponentu

import Navbar from '../Navbar/Navbar';

import Footer from '../Footer/Footer';

const ContactForm = ({messages,setMessages}) => {

  const [name, setName] = useState('');

  const [email, setEmail] = useState('');

  const [message, setMessage] = useState('');

  const [alert, setAlert] = useState('');

  const handleSubmit = (e) => {

    e.preventDefault();

    if (name && email && message) {

      const newMessage = {

        name,

        email,

        message,

      };

      setMessages([...messages, newMessage]);

      setAlert('Message sent successfully!');

      setName('');

      setEmail('');

      setMessage('');

    } else {

      setAlert('Please fill in all fields.');

    }

  };

  return (

    <><Navbar/>

    <div className="contact-form-container">

      <h2>Contact Us</h2>

      <form onSubmit={handleSubmit} className="contact-form">

        <div className="input-container">

          <label>Name:</label>

          <TextInput

            value={name}

            onChange={(e) => setName(e.target.value)}

            placeholder="Enter your name"

          />

        </div>

        <div className="input-container">

          <label>Email:</label>

          <TextInput

            type="email"

            value={email}

            onChange={(e) => setEmail(e.target.value)}

            placeholder="Enter your email"

          />

        </div>

        <div className="input-container">

          <label>Message:</label>

          <textarea

            value={message}

            onChange={(e) => setMessage(e.target.value)}

          ></textarea>

        </div>

        <button type="submit" className="submit-button">

          Submit

        </button>

      </form>

      {alert && <div className="alert">{alert}</div>}

    </div>

    <Footer></Footer>

    </>

  );

};

export default ContactForm;

import React from 'react';

const TextInput = ({ value, onChange, ...props }) => {

  return (

    <input

      type="text"

      value={value}

      onChange={onChange}

      {...props}

    />

  );

};

export default TextInput;

Komponenta "Messages" predstavlja deo aplikacije koji omogućava pregled, upravljanje i brisanje poruka koje su korisnici poslali agenciji za nekretnine. Ova komponenta pruža tabelarni prikaz svih primljenih poruka, uključujući informacije o imenu pošiljaoca, email adresi i sadržaju poruke. Korisnicima je takođe omogućeno da iz tabele izbrišu određene poruke pritiskom na dugme "Delete".

import React from 'react';

import './Messages.css';

import Footer from '../Footer/Footer';

import Navbar from '../Navbar/Navbar';

const Messages = ({ messages, setMessages }) => {

  const handleDelete = (index) => {

    const updatedMessages = [...messages];

    updatedMessages.splice(index, 1);

    setMessages(updatedMessages);

  };

  return (

    <>

    <Navbar></Navbar>

    <div className="messages-container">

      <h2>Messages</h2>

      <table className="messages-table">

        <thead>

          <tr>

            <th>Name</th>

            <th>Email</th>

            <th>Message</th>

            <th>Action</th>

          </tr>

        </thead>

        <tbody>

          {messages.map((message, index) => (

            <tr key={index}>

              <td>{message.name}</td>

              <td>{message.email}</td>

              <td>{message.message}</td>

              <td>

                <button onClick={() => handleDelete(index)}>Delete</button>

              </td>

            </tr>

          ))}

        </tbody>

      </table>

    </div>

    <Footer></Footer>

    </>

  );

};

export default Messages;

Komponenta "PropertyList" predstavlja ključni deo aplikacije za pregled nekretnina. Ova komponenta omogućava korisnicima da pretražuju i filtriraju različite nekretnine na osnovu kriterijuma kao što su cena, tip nekretnine i ključne reči u opisu ili naslovu. Pored toga, komponenta omogućava paginaciju rezultata kako bi korisnicima bilo lakše pregledati veliki broj nekretnina. Takođe, koristi se i custom hook useNekretnine za dobijanje podataka o nekretninama sa servera. Svaka nekretnina se prikazuje kroz komponentu "PropertyCard", gde se prikazuju osnovne informacije o nekretnini, kao što su naslov, opis, cena, broj spavaćih soba i tip nekretnine.

import React, { useState, useEffect } from 'react';

import axios from 'axios';

import './PropertyList.css';

import PropertyCard from './PropertyCard';

import ReactPaginate from 'react-paginate';

import Navbar from '../Navbar/Navbar';

import Footer from '../Footer/Footer';

import useNekretnine from '../customHooks/useNekretnine';

const PropertyList = () => {

    const { data: properties, isLoading, error } = useNekretnine('http://127.0.0.1:8000/api/properties');

  const [searchTerm, setSearchTerm] = useState('');

  const [selectedPropertyType, setSelectedPropertyType] = useState('');

  const [minPrice, setMinPrice] = useState('');

  const [maxPrice, setMaxPrice] = useState('');

  const [filteredProperties, setFilteredProperties] = useState([]);

  const [currentPage, setCurrentPage] = useState(0);

  const propertiesPerPage = 3;

  useEffect(() => {

    // Funkcija za filtriranje po ceni

    const filterByPrice = () => {

      const filteredByPrice = properties.filter((property) =>

        (minPrice === '' || parseInt(property.price) >= parseInt(minPrice)) &&

        (maxPrice === '' || parseInt(property.price) <= parseInt(maxPrice))

      );

      applyFilters(filteredByPrice);

    };

    filterByPrice();

  }, [minPrice, maxPrice, properties]);

  const applyFilters = (filteredData) => {

    // Funkcija za primenu svih filtera

    const filteredBySearch = filteredData.filter((property) =>

      property.description.toLowerCase().includes(searchTerm.toLowerCase()) ||

      property.title.toLowerCase().includes(searchTerm.toLowerCase())

    );

    if (selectedPropertyType) {

      const filteredByType = filteredBySearch.filter(

        (property) => property.propery\_type.name === selectedPropertyType

      );

      setFilteredProperties(filteredByType);

    } else {

      setFilteredProperties(filteredBySearch);

    }

    /////

  };

  const filterBySearchTerm = (value) => {

    setSearchTerm(value);

    applyFilters(properties);

  };

  const filterByPropertyType = (type) => {

    setSelectedPropertyType(type);

    applyFilters(properties);

  };

  const handlePageChange = ({ selected }) => {

    setCurrentPage(selected);

  };

  const pageCount = Math.ceil(filteredProperties.length / propertiesPerPage);

  const offset = currentPage \* propertiesPerPage;

  const currentProperties = filteredProperties.slice(offset, offset + propertiesPerPage);

  return (

    <>

    <Navbar></Navbar>

    <div className="property-list">

      <div className="search-container">

        <input

          type="text"

          placeholder="Search by description or title"

          value={searchTerm}

          onChange={(e) => filterBySearchTerm(e.target.value)}

        />

        <select

          value={selectedPropertyType}

          onChange={(e) => filterByPropertyType(e.target.value)}

        >

          <option value="">All Types</option>

          <option value="Stan">Stan</option>

          <option value="Kuća">Kuća</option>

          <option value="Apartman">Apartman</option>

          <option value="Vikendica">Vikendica</option>

          <option value="Poslovni prostor">Poslovni prostor</option>

        </select>

        <input

          type="number"

          placeholder="Min Price"

          value={minPrice}

          onChange={(e) => setMinPrice(e.target.value)}

        />

        <input

          type="number"

          placeholder="Max Price"

          value={maxPrice}

          onChange={(e) => setMaxPrice(e.target.value)}

        />

      </div>

      {isLoading ? (

        <div>Loading...</div>

      ) : (

        <>

          <ReactPaginate

            previousLabel={'Previous'}

            nextLabel={'Next'}

            breakLabel={'...'}

            pageCount={pageCount}

            marginPagesDisplayed={2}

            pageRangeDisplayed={5}

            onPageChange={handlePageChange}

            containerClassName={'pagination'}

            activeClassName={'active'}

          />

          {currentProperties.map((property) => (

            <PropertyCard key={property.id} property={property}></PropertyCard>

          ))}

        </>

      )}

    </div>

    <Footer></Footer>

    </>

  );

};

export default PropertyList;

import React from 'react';

import './PropertyCard.css';

const PropertyCard = ({ property }) => {

  return (

    <div className="property-card">

      <div className="property-header">

        <h3>{property.title}</h3>

      </div>

      <div className="property-content">

        <p>Description: {property.description}</p>

        <p>Price: {property.price}</p>

        <p>Bedrooms: {property.bedrooms}</p>

        <p>Property Type: {property.propery\_type.name}</p>

      </div>

    </div>

  );

};

export default PropertyCard;

# Korisničko uputstvo

Prilikom pokretanja aplikacije otvara se početna stranica kao na slici ispod

Slika na kojoj se nalazi tekst, snimak ekrana, Font, dizajn

Opis je automatski generisan

Odabirom opcije PropertyList iz navigacionog menija, otvara se stranica na kojoj korisnici mogu da pretražuju i filtriraju nekretnine. Na ovoj stranici je takođe implementirana i paginacija

Slika na kojoj se nalazi tekst, softver, broj, Ikona na računaru

Opis je automatski generisan

Na stranici Contact se nalazi kontakt forma

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Veb stranica

Opis je automatski generisan

Na komponenti Messages se nalazi tabela na kojoj su prikazane sve poruke koje su korisnici poslali kroz kontakt formu

Slika na kojoj se nalazi tekst, softver, Ikona na računaru, Veb stranica

Opis je automatski generisan