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

Centralni deo aplikacije je posvećen pregledu, dodavanju, ažuriranju i brisanju usluga koje su u poduni u sistemu euprave, omogućavajući korisnicima da efikasno upravljaju informacijama o različitim uslugama. Korisnici mogu pregledati sve dostupne usluge, filtrirati ih prema određenim kriterijumima kao što su naziv, kategorija i prioritet, te pristupiti detaljnim informacijama o svakoj usluzi. Funkcionalnost za dodavanje novih usluga omogućava unos svih relevantnih podataka o usluzi, uključujući naziv, opis, cenu, kategoriju, kao i druge važne informacije. Postoji i mogućnost ažuriranja postojećih usluga, gde korisnici mogu izmeniti već unete podatke.

Pored upravljanja uslugama, aplikacija sadrži i početnu stranicu koja pruža brz pregled najvažnijih informacija i usluga dostupnih u aplikaciji, sa uključenim slikama i relevantnim detaljima. Intuitivni navigacioni meni olakšava kretanje kroz aplikaciju, omogućavajući korisnicima brz pristup svim ključnim delovima sistema.

# Implementacija

U ovoj React aplikaciji, implementirani su koncepti rutiranja koristeći react-router-dom biblioteku, koja omogućava navigaciju između različitih komponenata bez potrebe za ponovnim učitavanjem cele stranice. Glavna komponenta App koristi BrowserRouter za obuhvatanje celokupne aplikacije, omogućavajući rutiranje na klijentskoj strani. Unutar BrowserRouter, komponenta Navbar pruža navigacione linkove, koristeći Link komponentu iz react-router-dom za kreiranje veza ka različitim stranicama unutar aplikacije, poput početne stranice, stranice za dodavanje usluga, pregleda usluga i stranice za ažuriranje usluga. Routes komponenta definiše različite rute (Route), svaka sa svojom putanjom (path) i elementom koji treba renderovati, povezujući URL putanje sa odgovarajućim React komponentama kao što su PocetnaStranica, Usluge, NovaUslugaForma i AzurirajUslugu. Ovaj pristup rutiranju omogućava efikasnu organizaciju i navigaciju unutar aplikacije, uz lako upravljanje sadržajem koji se prikazuje korisniku u zavisnosti od trenutne URL putanje.

import React from 'react';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import PocetnaStranica from './components/home/Pocetna';

import Usluge from './components/usluge/Usluge';

import Navbar from './components/navbar/Navbar';

import NovaUslugaForma from './components/usluge/NovaUslugaForma';

import AzurirajUslugu from './components/usluge/AzurirajUslugu';

function App() {

  return (

    <BrowserRouter>

    <Navbar></Navbar>

      <Routes>

        <Route path="/" element={<PocetnaStranica />} />

        <Route path="usluge/promeni/:id" element={<AzurirajUslugu />} />

        <Route path="/usluge/dodaj" element={<NovaUslugaForma />} />

        <Route path="/usluge" element={<Usluge />} />

      </Routes>

    </BrowserRouter>

  );

}

export default App;

import React from 'react';

import { Link } from 'react-router-dom';

import './Navbar.css';

function Navbar() {

  return (

    <nav className="navbar">

      <div className="nav-container">

        <div className="nav-links">

          <Link to="/" className="nav-item">Početna</Link>

          <Link to="/usluge/dodaj" className="nav-item">Dodaj</Link>

          <Link to="/usluge" className="nav-item">Usluge</Link>

        </div>

      </div>

    </nav>

  );

}

export default Navbar;

Komponenta PocetnaStranica u React aplikaciji predstavlja početnu stranicu sa ključnim informacijama i uslugama. Korišćenjem React Hooks (useState i useEffect), komponenta upravlja prikazom slika i sadržaja. U useState se čuvaju slike koje se dinamički učitavaju iz Unsplash API-ja. useEffect se koristi za postavljanje ovih slika prilikom inicijalnog učitavanja komponente. Komponenta takođe uključuje Slider iz react-slick biblioteke za prikaz slika u obliku karusela, sa definisanim postavkama kao što su automatsko prelistavanje, brzina prelaza i sl. U okviru stranice, prikazane su različite kategorije usluga kao što su porodični život, vrtići, obrazovanje i zapošljavanje, svaka sa svojim opisom i identifikatorom.

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

import Slider from 'react-slick';

import './Pocetna.css';

import 'slick-carousel/slick/slick.css';

import 'slick-carousel/slick/slick-theme.css';

function PocetnaStranica() {

  const [images, setImages] = useState([]);

  useEffect(() => {

    // Here you would fetch images from the Unsplash API

    // For demonstration, we're using static image URLs.

    setImages([

      'https://source.unsplash.com/featured/?school',

      'https://source.unsplash.com/featured/?kindergarten',

      'https://source.unsplash.com/featured/?job',

      'https://source.unsplash.com/featured/?education',

      'https://source.unsplash.com/featured/?family'

    ]);

  }, []);

  const settings = {

    dots: true,

    infinite: true,

    speed: 500,

    slidesToShow: 1,

    slidesToScroll: 1,

    autoplay: true,

    autoplaySpeed: 3000

  };

  return (

    <div className="pocetna">

      <header className="pocetna-header">

        <h1>Dobro došli</h1>

        <p>U par koraka do svih informacija i elektronskih usluga u Republici Srbiji.</p>

      </header>

      <section className="pocetna-services">

        <div className="service-card" id="family">

          <h2>Porodični život</h2>

          <p>Trudničko i porodiljsko bolovanje, naknade, prijava deteta, lečenje lopatak...</p>

        </div>

        <div className="service-card" id="birth">

          <h2>Vrtic</h2>

          <p>Upis u vrtic, eVrtic...</p>

        </div>

        <div className="service-card" id="education">

          <h2>Osnovno obrazovanje</h2>

          <p>Proceduri dovarak, dokumentacija za upis deteta u osnovnu školu, elektronski dnevnik...</p>

        </div>

        <div className="service-card" id="employment">

          <h2>Tražim posao</h2>

          <p>Najlakši načini do posla.</p>

        </div>

      </section>

      <Slider {...settings}>

          {images.map((img, index) => (

            <div key={index}>

              <img src={img} alt="Slika" />

            </div>

          ))}

        </Slider>

    </div>

  );

}

export default PocetnaStranica;

U komponenti Usluge, implementirana je funkcionalnost za prikaz, pretragu, brisanje i ažuriranje usluga koristeći React i axios za upravljanje HTTP zahtevima. Pomoću kuke useFetchServices, podaci o uslugama se dinamički učitavaju sa zadatog URL-a. Ova kuka upravlja stanjima kao što su loading i error za efikasno upravljanje procesom učitavanja i grešaka. Nakon što se podaci učitaju, korisnici mogu pretraživati usluge po različitim kriterijumima kao što su naziv, kategorija i prioritet. Komponenta takođe omogućava paginaciju rezultata, prikazujući ograničen broj usluga po stranici za bolju preglednost.

RedTabele je odgovorna za prikaz pojedinačnih redova u tabeli usluga. Ova komponenta omogućava brisanje usluga putem axios DELETE zahteva i navigaciju do stranice za ažuriranje usluga, gde korisnici mogu izmeniti postojeće informacije. Svaki red u tabeli sadrži informacije o usluzi kao što su naziv, opis, cena, kategorija, i druge relevantne informacije, uz opcije za brisanje i ažuriranje.

import { useState, useEffect } from 'react';

import axios from 'axios';

const useFetchServices = (url) => {

  const [data, setData] = useState([]);

  const [loading, setLoading] = useState(true);

  const [error, setError] = useState(null);

  useEffect(() => {

    setLoading(true);

    axios.get(url)

      .then(response => {

        setData(response.data);

        setLoading(false);

      })

      .catch(error => {

        setError(error);

        setLoading(false);

      });

  }, [url]);

  return { data, setData,loading, error };

};

export default useFetchServices;

import React from 'react';

import axios from 'axios';

import { useNavigate } from 'react-router-dom';

function RedTabele({ usluga, setUsluge }) {

  const navigate = useNavigate();

  const handleDelete = (id) => {

    if (window.confirm("Da li ste sigurni da želite da obrišete uslugu?")) {

      axios.delete(`http://127.0.0.1:8000/api/usluge/${id}`)

        .then(response => {

          alert('Usluga je uspešno obrisana.');

          setUsluge((prevUsluge) => prevUsluge.filter((usluga) => usluga.id !== id));

        })

        .catch(error => {

          console.error('There was an error deleting the usluga!', error);

        });

    }

  };

  const goToUpdatePage = (id) => {

    navigate(`promeni/${id}`);

  };

  return (

    <tr>

      <td>{usluga.naziv}</td>

      <td>{usluga.opis}</td>

      <td>{usluga.cena}</td>

      <td>{usluga.kategorija}</td>

      <td>{usluga.vreme\_obrade}</td>

      <td>{usluga.dokumentacija\_potrebna}</td>

      <td>{usluga.digitalni\_potpis\_potreban ? 'Da' : 'Ne'}</td>

      <td>{usluga.prioritet}</td>

      <td>

        <button onClick={() => goToUpdatePage(usluga.id)}>Promeni</button>

        <button onClick={() => handleDelete(usluga.id)}>Obriši</button>

      </td>

    </tr>

  );

}

export default RedTabele;

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

import axios from 'axios';

import './Usluge.css';

import RedTabele from './RedTabele';

import useFetchServices from './useFetchServices ';

function Usluge() {

  const { data: usluge,setData:setUsluge, loading, error } = useFetchServices('http://127.0.0.1:8000/api/usluge');

  const [filteredUsluge, setFilteredUsluge] = useState([]);

  const [search, setSearch] = useState({

    naziv: '',

    kategorija: '',

    prioritet: '',

  });

  const [currentPage, setCurrentPage] = useState(1);

  const servicesPerPage = 5;

  useEffect(() => {

    setFilteredUsluge(usluge);

  }, [usluge]);

  useEffect(() => {

    const filteredData = usluge.filter(usluga =>

      (!search.naziv || usluga.naziv.toLowerCase().includes(search.naziv.toLowerCase())) &&

      (!search.kategorija || usluga.kategorija.toLowerCase().includes(search.kategorija.toLowerCase())) &&

      (!search.prioritet || usluga.prioritet.toLowerCase().includes(search.prioritet.toLowerCase()))

    );

    setFilteredUsluge(filteredData);

  }, [search, usluge]);

  const handleSearchChange = (e) => {

    const { name, value } = e.target;

    setSearch({ ...search, [name]: value });

    setCurrentPage(1); // Reset to first page when search changes

  };

  const handleSearch = (e) => {

    e.preventDefault();

  };

  // Get current services

  const indexOfLastService = currentPage \* servicesPerPage;

  const indexOfFirstService = indexOfLastService - servicesPerPage;

  const currentServices = filteredUsluge.slice(indexOfFirstService, indexOfLastService);

  const totalPages = Math.ceil(filteredUsluge.length / servicesPerPage);

  // Change page

  const paginate = (pageNumber) => setCurrentPage(pageNumber);

  if (loading) return <p>Loading...</p>;

  if (error) return <p>Error loading services: {error.message}</p>;

  return (

    <div className="usluge">

    <form onSubmit={handleSearch}>

      <input

          type="text"

          name="naziv"

          value={search.naziv}

          onChange={handleSearchChange}

          placeholder="Pretraži po nazivu"

        />

        <input

          type="text"

          name="kategorija"

          value={search.kategorija}

          onChange={handleSearchChange}

          placeholder="Pretraži po kategoriji"

        />

        <select name="prioritet" value={search.prioritet} onChange={handleSearchChange}>

          <option value="">Svi prioriteti</option>

          <option value="low">Nizak</option>

          <option value="medium">Srednji</option>

          <option value="high">Visok</option>

        </select>

      <button type="submit">Pretraži</button></form>

      <table>

        <thead>

          <tr>

            <th>Naziv</th>

            <th>Opis</th>

            <th>Cena</th>

            <th>Kategorija</th>

            <th>Vreme Obrade</th>

            <th>Dokumentacija Potrebna</th>

            <th>Digitalni Potpis Potreban</th>

            <th>Prioritet</th>

            <th>Obrisi</th>

          </tr>

        </thead>

        <tbody>

          {currentServices.map(usluga => (

            <RedTabele key={usluga.id} usluga={usluga} setUsluge={setUsluge} />

          ))}

        </tbody>

      </table>

      <div className="pagination">

        {Array.from({ length: totalPages }, (\_, index) => (

          <button

            key={index}

            onClick={() => paginate(index + 1)}

            className={currentPage === index + 1 ? 'active' : ''}

          >

            {index + 1}

          </button>

        ))}

      </div>

    </div>

  );

}

export default Usluge;

Komponenta NovaUslugaForma u React aplikaciji služi za dodavanje novih usluga kroz formu koja komunicira sa back-end serverom putem axios HTTP biblioteke. Korisnik može uneti podatke o usluzi, uključujući naziv, opis, cenu, kategoriju, vreme obrade, potrebnu dokumentaciju, informaciju da li je digitalni potpis potreban, kao i prioritet usluge. Kada se forma popuni i potvrdi, podaci se šalju back-endu putem POST zahteva na definisani URL. Stanje forme se upravlja korišćenjem React hook-a useState, a promene u input poljima se obrađuju putem handleInputChange funkcije, koja ažurira stanje forme. Po uspešnom dodavanju usluge, korisnik dobija obaveštenje, a forma se resetuje za eventualno dodavanje sledeće usluge. U slučaju greške pri slanju podataka, korisnik će biti obavešten o grešci.

import React, { useState } from 'react';

import axios from 'axios';

import './NovaUslugaForma.css';

function NovaUslugaForma() {

  const [formData, setFormData] = useState({

    naziv: '',

    opis: '',

    cena: '',

    kategorija: '',

    vreme\_obrade: '',

    dokumentacija\_potrebna: '',

    digitalni\_potpis\_potreban: false,

    prioritet: ''

  });

  const handleInputChange = (e) => {

    const { name, value, type, checked } = e.target;

    setFormData({

      ...formData,

      [name]: type === 'checkbox' ? checked : value

    });

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    axios.post('http://127.0.0.1:8000/api/usluge', formData)

      .then(response => {

        alert('Usluga je uspešno dodata!');

        // Optionally reset form or handle response further

        setFormData({

          naziv: '',

          opis: '',

          cena: '',

          kategorija: '',

          vreme\_obrade: '',

          dokumentacija\_potrebna: '',

          digitalni\_potpis\_potreban: false,

          prioritet: ''

        });

      })

      .catch(error => {

        console.error('There was an error submitting the form!', error);

      });

  };

  return (

    <form className="nova-usluga-forma" onSubmit={handleSubmit}>

      <input

        type="text"

        name="naziv"

        value={formData.naziv}

        onChange={handleInputChange}

        placeholder="Naziv usluge"

        required

      />

      <textarea

        name="opis"

        value={formData.opis}

        onChange={handleInputChange}

        placeholder="Opis usluge"

        required

      />

      <input

        type="number"

        name="cena"

        value={formData.cena}

        onChange={handleInputChange}

        placeholder="Cena usluge"

        required

      />

      <input

        type="text"

        name="kategorija"

        value={formData.kategorija}

        onChange={handleInputChange}

        placeholder="Kategorija usluge"

        required

      />

      <input

        type="number"

        name="vreme\_obrade"

        value={formData.vreme\_obrade}

        onChange={handleInputChange}

        placeholder="Vreme obrade (u danima)"

        required

      />

      <input

        type="text"

        name="dokumentacija\_potrebna"

        value={formData.dokumentacija\_potrebna}

        onChange={handleInputChange}

        placeholder="Dokumentacija potrebna"

      />

      <label>

        Digitalni potpis potreban:

        <input

          type="checkbox"

          name="digitalni\_potpis\_potreban"

          checked={formData.digitalni\_potpis\_potreban}

          onChange={handleInputChange}

        />

      </label>

      <select name="prioritet" value={formData.prioritet} onChange={handleInputChange} required>

        <option value="">Izaberite prioritet</option>

        <option value="low">Nizak</option>

        <option value="medium">Srednji</option>

        <option value="high">Visok</option>

      </select>

      <button type="submit">Dodaj Uslugu</button>

    </form>

  );

}

export default NovaUslugaForma;

Komponenta AzurirajUslugu u React aplikaciji omogućava ažuriranje podataka o postojećoj usluzi. Nakon što korisnik odabere uslugu za ažuriranje, komponenta koristi ID usluge iz URL-a (dobijenog preko useParams kuke) za učitavanje detalja te usluge pomoću axios GET zahteva. Inicijalno stanje forme, koje upravlja podacima usluge, postavlja se na osnovu odgovora dobijenog sa servera. Forma koristi InputField komponente za unos podataka kao što su naziv, opis, cena, kategorija, vreme obrade i dokumentacija potrebna. Takođe postoji checkbox za digitalni potpis i selekt meni za odabir prioriteta. Promene u input poljima se obrađuju kroz handleInputChange funkciju koja ažurira stanje forme. Kada se forma pošalje, izvršava se axios PUT zahtev na server sa ažuriranim podacima usluge. U slučaju uspeha, korisnik dobija obaveštenje o uspešnom ažuriranju, dok se u slučaju greške prikazuje odgovarajuća poruka o grešci.

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

import axios from 'axios';

import { useParams } from 'react-router-dom';

import './NovaUslugaForma.css';

import InputField from './InputField';

function AzurirajUslugu() {

  const [formData, setFormData] = useState({

    naziv: '',

    opis: '',

    cena: '',

    kategorija: '',

    vreme\_obrade: '',

    dokumentacija\_potrebna: '',

    digitalni\_potpis\_potreban: false,

    prioritet: ''

  });

  const { id } = useParams();

  useEffect(() => {

    axios.get(`http://127.0.0.1:8000/api/usluge/${id}`)

      .then(response => {

        setFormData(response.data);

      })

      .catch(error => {

        console.error('There was an error fetching the service details!', error);

      });

  }, [id]);

  const handleInputChange = (e) => {

    const { name, value, type, checked } = e.target;

    setFormData({

      ...formData,

      [name]: type === 'checkbox' ? checked : value

    });

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    axios.put(`http://127.0.0.1:8000/api/usluge/${id}`, formData)

      .then(response => {

        alert('Usluga je uspešno ažurirana!');

      })

      .catch(error => {

        console.error('There was an error updating the service!', error);

      });

  };

  return (

    <form className="nova-usluga-forma" onSubmit={handleSubmit}>

      <label htmlFor="naziv">Naziv usluge</label>

      <InputField

        label="Naziv usluge"

        type="text"

        name="naziv"

        value={formData.naziv}

        onChange={handleInputChange}

        required

      />

      <InputField

        label="Opis usluge"

        type="text"

        name="opis"

        value={formData.opis}

        onChange={handleInputChange}

        required

      />

      <InputField

        label="Cena usluge"

        type="number"

        name="cena"

        value={formData.cena}

        onChange={handleInputChange}

        required

      />

      <InputField

        label="Kategorija usluge"

        type="text"

        name="kategorija"

        value={formData.kategorija}

        onChange={handleInputChange}

        required

      />

      <InputField

        label="Vreme obrade (u danima)"

        type="number"

        name="vreme\_obrade"

        value={formData.vreme\_obrade}

        onChange={handleInputChange}

        required

      />

      <InputField

        label="Dokumentacija potrebna"

        type="text"

        name="dokumentacija\_potrebna"

        value={formData.dokumentacija\_potrebna}

        onChange={handleInputChange}

      />

      <label htmlFor="digitalni\_potpis\_potreban">

        Digitalni potpis potreban:

        <input

          id="digitalni\_potpis\_potreban"

          type="checkbox"

          name="digitalni\_potpis\_potreban"

          checked={formData.digitalni\_potpis\_potreban}

          onChange={handleInputChange}

        />

      </label>

      <label htmlFor="prioritet">Prioritet</label>

      <select

        id="prioritet"

        name="prioritet"

        value={formData.prioritet}

        onChange={handleInputChange}

        required

      >

        <option value="">Izaberite prioritet</option>

        <option value="low">Nizak</option>

        <option value="medium">Srednji</option>

        <option value="high">Visok</option>

      </select>

      <button type="submit">Ažuriraj Uslugu</button>

    </form>

  );

}

export default AzurirajUslugu;

# Korisničko uputstvo

Početna stranica na kojoj korisnici mogu da vide osnovni set usluga koji se nudi, kao i razne slike građana.

Slika na kojoj se nalazi tekst, nebo, drvo, snimak ekrana

Opis je automatski generisan

Stranica Usluge na kojoj korisnici mogu vrsiti filtritranje, brisanje ili izmenu usluga. Ovde je implemnetirana i paginacija.

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

Opis je automatski generisan

Klikom na dugme promeni otvara se forma za izmenu određene usluge

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Odabirom opcije Dodaj iz navigacionog menija, korisnicima se otvara forma za dodavanje novih usluge

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan