Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

ReactJS

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Nikola | Mečanin | 2019/0227 |
| Jovana | Bujak | 2019/1074 |
| Link ka Github-u | https://github.com/elab-development/internet-tehnologije-projekat-podacinainternetu\_2019\_0227.git | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc117932679)

[2 Implementacija 3](#_Toc117932680)

[3 Korisničko uputstvo 3](#_Toc117932681)

# Korisnički zahtev

Ova aplikacija omogućava korisnicima da se prijave koristeći svoje kredencijale i pristupe centralizovanom hubu gde mogu upravljati zadacima, zaposlenima, fajlovima i detaljima o firmama. Nakon prijave, korisnici se susreću sa početnom stranicom koja služi kao uvod u glavne funkcije aplikacije i pruža brz pristup ostalim delovima sistema.

Kroz aplikaciju, korisnici mogu pregledati listu firmi, gde svaka firma ima pridružene zadatke i fajlove, čime se olakšava organizacija i praćenje svakodnevnih operacija. Detalji firmi su lako dostupni, i prikazuju sve relevantne informacije koje su bitne za poslovanje. Za zaposlene unutar kompanije, aplikacija nudi pregled i upravljanje njihovim zadacima, uključujući mogućnost filtriranja zadataka na osnovu njihovog statusa, što pruža efikasan način praćenja napretka i prioriteta.

Pored toga, aplikacija omogućava upravljanje fajlovima kroz integraciju sa Dropbox servisom. Korisnici mogu lako otpremati nove fajlove i pregledati postojeće, pristupajući direktnim linkovima za otvaranje ili preuzimanje fajlova smeštenih na Dropbox-u. Svaki fajl je prikazan tabelarno sa osnovnim informacijama kao što su ime fajla, datum poslednje modifikacije i veličina, čime se omogućava brz i jasan pregled.

Za komunikaciju sa serverom koristi se RESTful API, gde svaki zahtev zahteva autentifikaciju korisnika, obezbeđujući tako sigurnost i privatnost podataka. Funkcionalnosti prijave i odjave su ključne za održavanje sigurnosti korisničkih sesija i podataka.

# Implementacija

U ovom poglavlju će biti dat kratak pregled najbitnijih delova koda

Komponenta LoginPage je dizajnirana da omogući korisnicima prijavljivanje u aplikaciju. Ona koristi stanje React-a (useState) za upravljanje korisničkim unosom e-pošte i lozinke, a takođe koristi axios za slanje asinhronog zahteva za prijavu na server. Nakon što korisnik unese svoje kredencijale i potvrdi formu, handleLogin funkcija obrađuje zahtev. U slučaju uspešne prijave, korisnički token se čuva u sessionStorage za dalju autentifikaciju tokom sesije, a korisnik se preusmerava na stranicu sa zadacima.

import React, { useState } from 'react';

import axios from 'axios';

import './LoginPage.css';

import InputField from './InputField';

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

const LoginPage = ({setToken}) => {

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

  const [password, setPassword] = useState('');

  let navigate = useNavigate();

  const handleLogin = async (event) => {

    event.preventDefault();

    try {

      const response = await axios.post('http://127.0.0.1:8000/api/login', {

        email,

        password,

      });

      const { access\_token } = response.data;

      sessionStorage.setItem('authToken', access\_token);

      sessionStorage.setItem('userId', response.data.user.id);

      setToken(access\_token);

      navigate('/tasks');

    } catch (error) {

      console.error('Login error', error);

    }

  };

  return (

    <div className="login-page">

      <form onSubmit={handleLogin} className="login-form">

        <InputField

            type="email"

            value={email}

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

            placeholder="Email"

            />

            <InputField

            type="password"

            value={password}

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

            placeholder="Password"

            />

        <button type="submit" className="login-button">Login</button>

      </form>

    </div>

  );

};

export default LoginPage;

komponenta Firme

Komponenta Firme služi a prikazivanje i pretraživanje informacija o firmama. Ona koristi React hooks (useState i useEffect) za upravljanje stanjem i asinhrono dohvatanje podataka o firmama putem axios biblioteke. Korisnici mogu pretraživati firme po nazivu ili PIB-u kroz tekstualno polje, a filtrirani rezultati se prikazuju tabelarno. Svaki red tabele, koji predstavlja pojedinačnu firmu, generiše se korišćenjem reusable komponente FirmaRow. Ova komponenta prikazuje osnovne informacije o firmi i omogućava navigaciju do detaljnijih podataka klikom na dugme "Detalji", koristeći useNavigate hook iz react-router-dom.

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

import axios from 'axios';

import './Firme.css';

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

import FirmaRow from './FirmaRow';

const Firme = () => {

  const [firme, setFirme] = useState([]);

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

  const navigate = useNavigate();

  useEffect(() => {

    const dohvatiFirme = async () => {

      try {

        const response = await axios.get('http://127.0.0.1:8000/api/firme');

        setFirme(response.data);

      } catch (error) {

        console.error('Došlo je do greške prilikom dohvatanja firmi', error);

      }

    };

    dohvatiFirme();

  }, []);

  const handleSearchChange = (e) => {

    setSearchTerm(e.target.value);

  };

  const filteredFirme = firme.filter(

    firma =>

      firma.naziv.toLowerCase().includes(searchTerm.toLowerCase()) ||

      firma.PIB.toLowerCase().includes(searchTerm.toLowerCase())

  );

  return (

    <div className="firme-container">

      <input

        type="text"

        placeholder="Pretraži po nazivu ili PIB-u"

        value={searchTerm}

        onChange={handleSearchChange}

        className="search-input"

      />

      <table className="firme-table">

        <thead>

          <tr>

            <th>ID</th>

            <th>Naziv</th>

            <th>PIB</th>

            <th>Detalji</th>

          </tr>

        </thead>

        <tbody>

          {filteredFirme.map((firma) => (

            <FirmaRow key={firma.id} firma={firma} onNavigate={navigate} />

          ))}

        </tbody>

      </table>

    </div>

  );

};

export default Firme;

Komponenta Tasks igra ključnu ulogu u prikazivanju i organizovanju zadataka (taskova) dodeljenih korisnicima. Ova komponenta efikasno koristi prilagođenu React kuku useTasks za dohvatanje liste zadataka sa servera, uzimajući u obzir ID trenutno prijavljenog korisnika, čuvanog u sessionStorage. Kada se učita, komponenta Tasks dinamički prikazuje zadatke koristeći tabelarni format, gde svaki red tabele, kreiran kroz TableRow reusable komponentu, sadrži detalje kao što su ID zadatka, njegov naziv, opis, rok za završetak i trenutni status.

Dodatna funkcionalnost komponente uključuje mogućnost filtriranja zadataka na osnovu njihovog statusa, kao što su "završeno", "otkazano" i "u izradi". Ovaj filter omogućava korisnicima da brzo pronađu i fokusiraju se na zadatke koji su relevantni za njihov trenutni radni tok. Selekcija filtera se vrši kroz padajući meni, a filtrirani zadaci se odmah prikazuju u tabeli, čime se postiže interaktivnost i veća korisnička praktičnost.

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

import axios from 'axios';

import './Tasks.css';

import useTasks from './useTasks';

import TableRow from './TableRow';

const Tasks = () => {

    const userId = parseInt(sessionStorage.getItem('userId'));

    const tasks = useTasks('http://127.0.0.1:8000/api/task', userId);

    const [filter, setFilter] = useState('sve');

    const filteredTasks = filter === 'sve' ? tasks : tasks.filter(task => task.status === filter);

  return (

    <div className="tasks-container">

         <div className="filter-container">

            <label htmlFor="statusFilter">Filter po statusu: </label>

            <select id="statusFilter" onChange={(e) => setFilter(e.target.value)} value={filter}>

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

            <option value="zavrseno">Završeno</option>

            <option value="otkazano">Otkazano</option>

            <option value="u izradi">U izradi</option>

            </select>

        </div>

      <table className="tasks-table">

        <thead>

          <tr>

            <th>ID</th>

            <th>Naziv</th>

            <th>Opis</th>

            <th>Rok</th>

            <th>Status</th>

          </tr>

        </thead>

        <tbody>

          {filteredTasks.map(task => (

            <TableRow key={task.id} task = {task}/>

          ))}

        </tbody>

      </table>

    </div>

  );

};

export default Tasks;

import { useState, useEffect } from 'react';

import axios from 'axios';

const useTasks = (url, userId) => {

  const [tasks, setTasks] = useState([]);

  useEffect(() => {

    const fetchTasks = async () => {

      try {

        const response = await axios.get(url, {

          headers: {

            Authorization: `Bearer ${sessionStorage.getItem('authToken')}`

          }

        });

        const filteredTasks = response.data.data.filter(task => task.zaposleni.id === userId);

        setTasks(filteredTasks);

      } catch (error) {

        console.error('Greška prilikom dohvatanja taskova', error);

      }

    };

    fetchTasks();

  }, [url, userId]);

  return tasks;

};

export default useTasks;

# Korisničko uputstvo

Prilikom pokretanja aplikacije otvara se početna stranica kao na slici ispod

Slika na kojoj se nalazi tekst, snimak ekrana, Operativni sistem, Font

Opis je automatski generisan

Odabriom opcije Prijava iz navigacionog menija, korisniku se otvara stranica koja sadrži login formu

Slika na kojoj se nalazi snimak ekrana, tekst, dizajn

Opis je automatski generisan

Kada se korisnik prijavi otvara mu se stranica na kojoj može da vidi i filtrira taskove koji su namenjeni za njega

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

Opis je automatski generisan

Odabirom opcije Firme iz navigacionog menija, korisniku se otvara tabelarni pregled svih firmi

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

Opis je automatski generisan

Na ovoj stranici korisnik može da pretražuje firme po nazivu ili PIBu, a klikom na Dugme detalji se prikazuju svi detalji o toj firmi

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

Opis je automatski generisan

Odabirom opcije fajlovi korisniku se otvara stranica na kojoj može da pregleda sve fajlove koji se nalaze na dropbox nalogu firme

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

Opis je automatski generisan

S obrzirom na to da firme imaju veliki broj fajlova, radi lakšeg snalaženja dodata je paginacija na tabelu.

Odabirom opcije Odjava, korisnici se odjavljuju iz aplikacije.