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 sadrži sistem za prikazivanje objava, gde se klikom na objavu mogu videti detaljniji podaci, uključujući komentare drugih korisnika. Korisnici mogu doprineti diskusiji dodavanjem svojih komentara, čime se podstiče dinamično okruženje za razgovor. Aplikacija obogaćuje interakciju korisnika omogućavajući sortiranje komentara po datumu, bilo u rastućem ili opadajućem redosledu, što korisnicima omogućava da pregledaju diskusije prema njihovoj relevantnosti ili svežini.

Aplikacija integriše funkciju za ažuriranje sadržaja u realnom vremenu, poput dnevnih citata i šala, pružajući osvežavajuće i angažujuće iskustvo svaki put kada korisnik poseti. Mogućnost pretrage objava povećava praktičnost za korisnike, omogućavajući lako pristupanje temama koje ih interesuju. Sa korisničkim interfejsom prijateljskim prema korisnicima, aplikacija je dizajnirana da podstiče angažovanje zajednice i učešće, čineći je svestranim alatom za online forume i diskusije. Uključivanje funkcija kao što su reakcije na objave i sortiranje komentara pozicionira je kao intuitivnu i odzivnu platformu za moderne online diskusije.

# Implementacija

Komponenta ForumHomePage koja služi kao početna stranica foruma. U ovoj komponenti koriste se React hookovi useState i useEffect za upravljanje stanjem i izvršavanje sporednih efekata. Dva stanja, quote i joke, inicijalizovana su kao prazni stringovi i koriste se za čuvanje citata dana i šale dana. U useEffect hooku, definisane su dve asinhrone funkcije, fetchQuote i fetchJoke, koje koriste Axios za dohvatanje podataka sa dva različita API-ja. fetchQuote dohvata slučajni citat sa quotable.io API-ja, dok fetchJoke dohvata šalu sa jokeapi.dev. Ovi podaci se zatim postavljaju u stanja quote i joke korišćenjem set funkcija. Struktura komponente uključuje HTML i JSX elemente za prikaz citata, šale i pravila ponašanja na forumu, a sve je stilizovano korišćenjem CSS-a definisanog u pocetna.css fajlu. Cilj ove komponente je da pruži korisnicima toplo dobrodošlicu na forum, uz zanimljiv sadržaj i važne informacije o pravilima ponašanja.

import axios from 'axios';

import './pocetna.css';

const ForumHomePage = () => {

  const [quote, setQuote] = useState('');

  const [joke, setJoke] = useState('');

  useEffect(() => {

    // Fetch a random quote

    const fetchQuote = async () => {

      try {

        const result = await axios.get('https://api.quotable.io/random');

        setQuote(result.data.content);

      } catch (error) {

        console.error('Error fetching the quote:', error);

      }

    };

    // Fetch a random joke

    const fetchJoke = async () => {

      try {

        const result = await axios.get('https://v2.jokeapi.dev/joke/Any');

        if (result.data.type === 'single') {

          setJoke(result.data.joke);

        } else {

          setJoke(`${result.data.setup} ... ${result.data.delivery}`);

        }

      } catch (error) {

        console.error('Error fetching the joke:', error);

      }

    };

    fetchQuote();

    fetchJoke();

  }, []);

  return (

    <div className="forum-homepage">

      <header className="header">

        <h1>Welcome to the Forum!</h1>

        <p>Join the conversation and share your thoughts!</p>

        <div className="icons">

          <i className="fas fa-comments"></i> {/\* Chat icon \*/}

          <i className="fas fa-smile"></i> {/\* Smile icon \*/}

          <i className="fas fa-users"></i> {/\* Users icon \*/}

        </div>

      </header>

      <div className="content">

        <div className="quote">

          <h2>Quote of the Day</h2>

          <p>{quote}</p>

        </div>

        <div className="joke">

          <h2>Joke of the Day</h2>

          <p>{joke}</p>

        </div>

      </div>

      <div className="rules-section">

        <h2>Pravila Ponašanja na Forumu</h2>

        <p>Dobrodošli u našu zajednicu! Molimo vas da se pridržavate sledećih pravila:</p>

        <ul>

          <li>Bez vređanja ili omalovažavanja drugih korisnika.</li>

          <li>Nema širenja govora mržnje ili diskriminacije bilo koje vrste.</li>

          <li>Budite ljubazni i poštujte mišljenja drugih.</li>

          <li>Ne širite lažne informacije ili sadržaje koji krše autorska prava.</li>

        </ul>

        <p>Zajedno možemo stvoriti prijateljsko i podržavajuće okruženje!</p>

      </div>

    </div>

  );

};

export default ForumHomePage;

Kod ispod predstavlja dve React komponente, ObjaveComponent i OjavaComponent, koje su deo forum aplikacije. ObjaveComponent je glavna komponenta koja koristi prilagođenu React kuku useObjave za dohvatanje liste objava (postova) sa servera. U ovoj komponenti implementirane su funkcionalnosti pretrage i paginacije. Korisnici mogu pretraživati objave po naslovu ili tekstu kroz input polje, a rezultati su paginirani, tj. prikazani na više stranica. Svaka objava prikazana je korišćenjem ObjavaComponent, koja prikazuje pojedinačnu objavu sa naslovom i tekstom, i sadrži dugme koje korisnike vodi na detaljniji prikaz te objave. ObjavaComponent koristi React Router useNavigate kuku za navigaciju do stranice sa detaljima objave. Obe komponente su stilizovane korišćenjem CSS-a, doprinoseći preglednosti i korisničkom iskustvu na forumu.

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

import axios from 'axios';

import ObjavaComponent from './ObjavaComponent';

import './ObjaveComponent.css';

import useObjave from './useObjave';

const ObjaveComponent = () => {

    const [objave, setObjave] = useObjave('http://127.0.0.1:8000/api/objave');

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

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

  const [itemsPerPage, setItemsPerPage] = useState(5);

  useEffect(() => {

    axios.get('http://127.0.0.1:8000/api/objave')

      .then(response => {

        setObjave(response.data.data);

      })

      .catch(error => console.error('Error fetching data:', error));

  }, []);

  const handleSearchChange = (e) => {

    setSearchTerm(e.target.value);

    setCurrentPage(1); // Reset to first page on search

  };

  const filteredObjave = objave.filter(objava =>

    objava.naziv.toLowerCase().includes(searchTerm.toLowerCase())

    || objava.tekst.toLowerCase().includes(searchTerm.toLowerCase())

  );

  // Pagination logic

  const indexOfLastItem = currentPage \* itemsPerPage;

  const indexOfFirstItem = indexOfLastItem - itemsPerPage;

  const currentItems = filteredObjave.slice(indexOfFirstItem, indexOfLastItem);

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

  const pageNumbers = [];

  for (let i = 1; i <= Math.ceil(filteredObjave.length / itemsPerPage); i++) {

    pageNumbers.push(i);

  }

  return (

    <div className="objave-container">

      <input

        type="text"

        placeholder="Pretraži objave..."

        value={searchTerm}

        onChange={handleSearchChange}

        className="search-input"

      />

      {currentItems.map(objava => (

        <ObjavaComponent key={objava.id} objava={objava} />

      ))}

      <div className="pagination">

        {pageNumbers.map(number => (

          <button key={number} onClick={() => paginate(number)} className="page-number">

            {number}

          </button>

        ))}

      </div>

    </div>

  );

};

export default ObjaveComponent;

import React from 'react';

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

const OjavaComponent = ({ objava }) => {

  const navigate = useNavigate();

  const goToDetails = () => {

    navigate(`/objave/${objava.id}`);

  };

  return (

    <div className="objava">

      <h2>{objava.naziv}</h2>

      <p>{objava.tekst}</p>

      <button onClick={goToDetails} className="details-button">DETALJI</button>

    </div>

  );

};

export default OjavaComponent;

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

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

import axios from 'axios';

import './Details.css';

import { FaThumbsUp, FaThumbsDown } from "react-icons/fa";

import CommentComponent from './CommentComponent ';

const Details = ({comments, setComments}) => {

  const { id } = useParams();

  const [objava, setObjava] = useState(null);

  const [newComment, setNewComment] = useState('');

  const [sortAscending, setSortAscending] = useState(true);

  useEffect(() => {

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

      .then(response => {

        setObjava(response.data.data);

      })

      .catch(error => console.error('Error fetching data:', error));

  }, [id]);

  const handleNewCommentChange = (e) => {

    setNewComment(e.target.value);

  };

Kod ispod se sastoji od dve React komponente, Details i CommentComponent, koje zajedno pružaju funkcionalnost za prikaz detalja o pojedinačnoj objavi na forumu i upravljanje komentarima. Komponenta Details koristi useParams iz React Router-a da dobije ID objave, a zatim koristi Axios za dohvatanje detalja te objave sa servera. U ovoj komponenti, korisnici mogu dodavati nove komentare, koji se zatim dodaju u globalni niz komentara. Postoji i mogućnost sortiranja komentara po datumu, uz opciju promene redosleda sortiranja. CommentComponent je zadužena za formatiranje i prikaz pojedinačnih komentara. Ona formatira datum komentara u čitljiviji format i prikazuje tekst komentara. Ove komponente zajedno omogućavaju korisnicima da pregledaju detalje objave, učestvuju u diskusiji dodavanjem komentara i pregledaju komentare na organizovan i intuitivan način.

  const submitComment = () => {

    const comment = {

      text: newComment,

      date: new Date().toISOString(),

      objavaId: id

    };

    setComments([...comments, comment]);

    setNewComment(''); // Reset the comment input field

  };

  if (!objava) {

    return <div>Loading...</div>;

  }

  const toggleSortOrder = () => {

    setSortAscending(!sortAscending);

  };

  // Function to sort comments based on date

  const sortedComments = [...comments].sort((a, b) => {

    const dateA = new Date(a.date);

    const dateB = new Date(b.date);

    return sortAscending ? dateA - dateB : dateB - dateA;

  });

  if (!objava) {

    return <div>Loading...</div>;

  }

  return (

    <div className="details-container">

           <div className="details-header" style={{ backgroundColor: '#yourColor' }}>

        <h2 className="details-title">{objava.naziv}</h2>

        <span className="details-category">{objava.tema.naziv\_teme}</span>

        <p className="details-date">{objava.datum\_objave}</p>

      </div>

      <p className="details-text">{objava.tekst}</p>

      <p className="details-info">

        <strong>Korisnik:</strong> {objava.user.ime}

      </p>

      <p className="details-info">

        <strong>Opis teme:</strong> {objava.tema.opis}

      </p>

      <div className="details-likes">

        <strong>

          <FaThumbsUp style={{ color: 'green' }} />

          {objava.broj\_svidjanja}

        </strong>

      </div>

      <div className="details-dislikes">

        <strong>

          <FaThumbsDown style={{ color: 'red' }} />

          {objava.broj\_nesvidjanja}

        </strong>

      </div>

      <div>

        <input

          type="text"

          placeholder="Dodaj komentar..."

          value={newComment}

          onChange={handleNewCommentChange}

        />

        <button onClick={submitComment}>Postavi komentar</button>

      </div>

      <button onClick={toggleSortOrder}>

        Sortiraj Komentare {sortAscending ? 'Opadajuće' : 'Rastuće'}

      </button>

      <div>

        {sortedComments.map((comment, index) => (

          <CommentComponent key={index} comment={comment} />

        ))}

      </div>

    </div>

  );

};

export default Details;

import React from 'react';

const CommentComponent = ({ comment }) => {

  // Convert ISO string to Date object

  const dateObj = new Date(comment.date);

  // Format the date in a more readable format (e.g., "Jan 01, 2024, 12:00 PM")

  const formattedDate = dateObj.toLocaleString('en-US', {

    month: 'short', // "Jan", "Feb", etc.

    day: '2-digit', // "01", "02", etc.

    year: 'numeric',

    hour: '2-digit', // "01", "02", etc., based on a 12-hour clock

    minute: '2-digit', // "00", "01", etc.

    hour12: true // Use 12-hour format with AM/PM

  });

  return (

    <div className="comment">

      <span className="comment-date">{formattedDate}</span>

      <span className="comment-text">{comment.text}</span>

    </div>

  );

};

export default CommentComponent;

# Korisničko uputstvo

Nakon što se aplikacija pokrene, otvara se početna stranica na kojoj korisnici mogu da vide Citat i šalu dana. Ove dve stvari se učitavaju sa spoljnog apija.

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

Opis je automatski generisan

Odabirom opcije objave iz navigacionog menija, korisnicima se prikazuje stranica na kojoj su prikazane sve objave koje postoje u bazi podataka. Radi lakšeg snalaženja, korisnicima je omogućena pretraga ovih objava, kao i paginacija.

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

Opis je automatski generisan

Klikom na dugme Detalji, otvaraju se detalji o određenoj objavi. Na ovoj stranici korisnici mogu da vide komentare drugih korisnika, kao i da ostavljaju svoje komentare. Omogućena je i funkcionalnost sortiranja komentara po datumu objave, radi lakše preglednosti.

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

Opis je automatski generisan