import React, { useState } from 'react';

import './Login.css';

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

const Login = () => {

    const [username, setUsername] = useState('');

    const [userPassword, setUserPassword] = useState('');

    const handleSubmit = (event) => {

        event.preventDefault();

        if (username === "") {

            alert("User name must be filled out");

            return;

        }

        if (userPassword === "") {

            alert("Password must be filled out");

            return;

        }

*// Add login logic here*

        console.log('Submitted:', { username, userPassword });

    };

    return (

        <div className="auth-form">

            <form onSubmit={handleSubmit}>

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

                    <div className="auth-input-group">

                        <div className="auth-title"><strong>Login</strong></div>

                        <label htmlFor="auth-username"><b>User name</b></label>

                        <input

                            type="text"

                            placeholder="Enter User name"

                            name="username"

                            className="auth-input user" *// Changed class name here*

                            value={username}

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

                            required

                        />

                    </div>

                    <div className="auth-input-group">

                        <label htmlFor="auth-password"><b>Password</b></label>

                        <input

                            type="password"

                            placeholder="Enter Password"

                            name="userPassword"

                            className="auth-input password"

                            value={userPassword}

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

                            required

                        />

                    </div>

                    <div>

                         <Link to="Main/Home"> <button type="submit" className="auth-button">Login</button> </Link>

                         </div>

                    <div className="auth-checkbox-container">

                        <label>

                            <input type="checkbox" name="rememberMe"/> Remember me

                        </label>

                    </div>

                </div>

                <span className="auth-password-reset">

                    Forgot <a href="forget" className="auth-link-password">password?</a>

                    <a href="register" className="auth-link-create">Register</a>

                </span>

            </form>

        </div>

    );

};

export default Login;

*/\* Login.css \*/*

.auth-form {

    display: flex;

    justify-content: center; */\* Center the form horizontally \*/*

    align-items: center; */\* Center the form vertically \*/*

    height: 97vh; */\* Full height of the viewport \*/*

    background-color: #f2f2f2; */\* Optional: Adds a light background for better visual centering \*/*

    padding: 10px; */\* Add some padding to ensure the form doesn't touch the edges of smaller screens \*/*

}

.auth-form form {

    background-color: #fff;

    padding: 20px;

    border-radius: 5px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

    width: 100%; */\* Use full width available \*/*

    max-width: 320px; */\* Constrain the form to a maximum width \*/*

    height: 405.99px;

}

.auth-form-container {

    display: flex;

    flex-direction: column;

    align-items: flex-start; */\* Align child items to the left \*/*

    width: 100%; */\* Ensure the container uses the full width \*/*

}

.auth-input-group {

    width: 100%;

    margin-bottom: 16px; */\* Space between input groups \*/*

    text-align: left; */\* Align contents of the group to the left \*/*

}

.auth-input-group label {

    margin-bottom: 8px;

    display: block;

    text-align: left; */\* Align label text to the left \*/*

}

.auth-input {

    padding: 10px;

    margin: 8px 0; */\* Margin between input boxes \*/*

    display: block; */\* Ensure the inputs are block elements \*/*

    border: 1px solid #ccc;

    box-sizing: border-box;

    border-radius: 4px;

    text-align: left; */\* Ensure input text starts from the left \*/*

*/\* Removed width: calc(100% - 20px); to use specific widths for each input \*/*

}

*/\* Adding specific styles for username input \*/*

.auth-input.user {

    height: 40.9px; */\* Increased height for username input \*/*

    padding: 15px; */\* Increased padding for larger input \*/*

    font-size: 16px; */\* Larger font size for better readability \*/*

    width: 99.8%; */\* Reduced width \*/*

}

*/\* Adding specific styles for password input \*/*

.auth-input.password {

    height: 40.9px; */\* Increased height for password input \*/*

    padding: 12px; */\* Increased padding for larger input \*/*

    font-size: 14px; */\* Font size adjustment \*/*

    width: 99.8%; */\* Reduced width \*/*

}

.auth-button {

    background-color: #4CAF50;

    color: white;

    padding: 14px 20px;

    margin: 8px 0;

    border: none;

    cursor: pointer;

    width: 360%;

    border-radius: 4px;

    text-align: center; */\* Center text within the button \*/*

    font-size: 20px;

}

.auth-button:hover {

    opacity: 0.8;

}

.auth-checkbox-container {

    display: flex;

    align-items: center;

    margin-top: 8px;

}

.auth-checkbox-container label {

    text-align: left; */\* Align label text to the left \*/*

}

.auth-link-password, .auth-link-create {

    text-decoration: none;

    color: #2196F3;

}

.auth-link-password:hover, .auth-link-create:hover {

    text-decoration: underline;

}

.auth-link-create {

    float: right; */\* Align the 'Create your Account' link to the right \*/*

}

.auth-password-reset {

    display: block;

    margin-top: 16px;

    text-align: left;

}

.auth-title {

    font-size: 30px;

    text-align: center; */\* Center the title text \*/*

    width: 100%;

}

*/\* Add media query for responsiveness \*/*

@media (max-width: 600px) {

    .auth-form {

        padding: 20px;

    }

    .auth-form form {

        width: 100%;

        padding: 15px;

    }

    .auth-input.user, .auth-input.password {

        width: 100%; */\* Adjust width to full available space on small screens \*/*

    }

}

.auth-title{

    margin-top: 20px;

    margin-bottom: 30px;

    font-size: 35px;

}

Update/delete

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

import axios from 'axios';

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

import './EventPage.css';

const EventList = () => {

  const [events, setEvents] = useState([]);

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

  const navigate = useNavigate();

  useEffect(() => {

    fetchEvents();

  }, []);

  const fetchEvents = () => {

    setIsLoading(true);

    axios.get('http://localhost:8080/api/v1/event')

      .then(response => {

        setEvents(response.data);

        console.log(response.data)

        setIsLoading(false);

      })

      .catch(error => {

        console.error('Error fetching event data:', error);

        setIsLoading(false);

      });

  };

  const handleAddEvent = () => {

    navigate('/Main/upload\_event');

  };

  const handleUpdateEvent = (id) => {

    navigate(`/Main/upload\_event/${id}`);

  };

  const handleDeleteEvent = (id) => {

    axios.delete(`http://localhost:8080/api/v1/event/${id}`)

      .then(response => {

        setEvents(events.filter(event => event.id !== id));

      })

      .catch(error => {

        console.error('Error deleting event:', error);

      });

  };

  return (

    <div className="event-table-container">

      <h2>Event List</h2>

      <button className="add-event" onClick={handleAddEvent}>Upload event</button>

      {isLoading ? (

        <p>Loading...</p>

      ) : (

        <table className="event-table">

          <thead>

            <tr>

              <th>ID</th>

              <th>Title</th>

              <th>Date</th>

              <th>Time</th>

              <th>Location</th>

              <th>Description</th>

              <th>Actions</th>

            </tr>

          </thead>

          <tbody>

            {events.map(event => (

              <tr key={event.id}>

                <td>{event.id}</td>

                <td>{event.title}</td>

                <td>{event.date}</td>

                <td>{event.time}</td>

                <td>{event.location}</td>

                <td>{event.description}</td>

                <td>

                  <button onClick={() => handleUpdateEvent(event.id)}>Update</button>

                  <button onClick={() => handleDeleteEvent(event.id)}>Delete</button>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      )}

    </div>

  );

};

export default EventList;

Update

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

import axios from 'axios';

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

import './UploadEvent.css';

const EventForm = () => {

    const { id } = useParams();

    const navigate = useNavigate();

    const [event, setEvent] = useState({

        title: '',

        date: '',

        time: '',

        location: '',

        description: ''

    });

    const [errorMessage, setErrorMessage] = useState('');

    useEffect(() => {

        if (id) {

*// Fetch existing event data if id is present*

            axios.get(`http://localhost:8080/api/v1/event/${id}`)

                .then(response => {

                    setEvent(response.data);

                })

                .catch(error => {

                    console.error('Error fetching event data:', error);

                    setErrorMessage('Failed to load event data.');

                });

        }

    }, [id]);

    const handleChange = (e) => {

        const { name, value } = e.target;

        setEvent(prevState => ({ ...prevState, [name]: value }));

    };

    const handleSubmit = async (e) => {

        e.preventDefault();

*// Validate the form data*

        if (!event.title || !event.date || !event.time || !event.location || !event.description) {

            setErrorMessage('All fields are required.');

            return;

        }

        try {

            if (id) {

*// Update existing event*

                await axios.put(`http://localhost:8080/api/v1/event/${id}`, event, {

                    headers: {

                        'Content-Type': 'application/json',

                    },

                });

            } else {

*// Create new event*

                await axios.post(`http://localhost:8080/api/v1/event`, event, {

                    headers: {

                        'Content-Type': 'application/json',

                    },

                });

            }

*// Redirect to event list after successful operation*

            navigate('/Main/event');

        } catch (error) {

            console.error('Error submitting event:', error);

            setErrorMessage('Failed to submit event. Please check your input and try again.');

        }

    };

    return (

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

            <h2>{id ? 'Update Event' : 'Add Event'}</h2>

            {errorMessage && <p className="error-message">{errorMessage}</p>}

            <form onSubmit={handleSubmit}>

                <div className="form-group">

                    <label htmlFor="title">Event Title:</label>

                    <input

                        type="text"

                        id="title"

                        name="title"

                        value={event.title}

                        onChange={handleChange}

                        required

                    />

                </div>

                <div className="form-group">

                    <label htmlFor="date">Date:</label>

                    <input

                        type="date"

                        id="date"

                        name="date"

                        value={event.date}

                        onChange={handleChange}

                        required

                    />

                </div>

                <div className="form-group">

                    <label htmlFor="time">Time:</label>

                    <input

                        type="time"

                        id="time"

                        name="time"

                        value={event.time}

                        onChange={handleChange}

                        required

                    />

                </div>

                <div className="form-group">

                    <label htmlFor="location">Location:</label>

                    <input

                        type="text"

                        id="location"

                        name="location"

                        value={event.location}

                        onChange={handleChange}

                        required

                    />

                </div>

                <div className="form-group">

                    <label htmlFor="description">Description:</label>

                    <textarea

                        id="description"

                        name="description"

                        value={event.description}

                        onChange={handleChange}

                        required

                    ></textarea>

                </div>

                <div className="form-buttons">

                    <button type="submit" className="btn btn-success">{id ? 'Update' : 'Add'}</button>

                    <button type="button" className="btn btn-danger" onClick={() => navigate('/Main/event')}>Cancel</button>

                </div>

            </form>

        </div>

    );

};

export default EventForm;