import { useState } from 'react';

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

import './Login.css';

const Login = () => {

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

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

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

  const validateLogin = () => {

    // Basic validation - you should implement proper authentication logic

    if (username === 'example' && password === 'password') {

      setErrorMessage(''); // Clear error message

      alert('Login successful!'); // Replace with your actual login logic

    } else {

      setErrorMessage('Invalid username or password');

    }

  };

  return (

    <div className="lcontainer">

      <form>

        <h2>Login</h2>

        <llabel htmlFor="username">Username:</llabel>

        <input

          type="text"

          id="username"

          name="username"

          value={username}

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

          required

        />

        <llabel htmlFor="password">Password:</llabel>

        <input

          type="password"

          id="password"

          name="password"

          value={password}

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

          required

        />

        <lbutton type="button" onClick={validateLogin}>

          Login

        </lbutton>

      </form>

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

      <p>Dont have an account? </p>

        {/\* <Link to="/signup">Sign Up</Link> \*/}

    </div>

  );

};

export default Login

body {

    background-image: url('https://source.unsplash.com/1600x900/?waterfront'); /\* Replace with your desired background image URL \*/

    background-size: cover;

    background-position: center;

    background-repeat: no-repeat;

    margin: 0; /\* Remove default margin to cover the entire viewport \*/

  }

.lcontainer {

    background-color: #060000;

    padding: 40px;

    border-radius: 5px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    opacity: 90%;

    height: 300px;

  }

  form {

    display: flex;

    flex-direction: column;

    margin-bottom: 5px;

    margin-top:-28px;

  }

  llabel {

    margin-bottom: 17px;

  }

  input {

    padding: 8px;

    margin-bottom: 16px;

  }

  lbutton {

    padding: 10px;

    background-color: #007BFF;

    color: #fff;

    border: none;

    border-radius: 17px;

    cursor: pointer;

  }

  lbutton:hover {

    background-color: #22364b;

  }

  #error-message {

    color: red;

    margin-top: 10px;

  }

// SignUp.jsx

import { useState } from 'react';

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

import './SignUp.css';

const SignUp = () => {

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

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

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

  const [confirmPassword, setConfirmPassword] = useState('');

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

  const validateSignUp = () => {

    // Basic validation - you should implement proper registration logic

    if (password !== confirmPassword) {

      setErrorMessage('Passwords do not match');

    } else {

      setErrorMessage('');

      alert('Registration successful!'); // Replace with your actual registration logic

    }

  };

  return (

    <div className="scontainer">

      <s-form>

        <h2>Create Your Account</h2>

        <label htmlFor="username">Username:</label>

        <input

          type="text"

          id="username"

          name="username"

          value={username}

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

          required

        />

        <label htmlFor="email">Email:</label>

        <input

          type="email"

          id="email"

          name="email"

          value={email}

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

          required

        />

        <label htmlFor="password">Password:</label>

        <input

          type="password"

          id="password"

          name="password"

          value={password}

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

          required

        />

        <label htmlFor="confirmPassword">Confirm Password:</label>

        <input

          type="password"

          id="confirmPassword"

          name="confirmPassword"

          value={confirmPassword}

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

          required

        />

        <s-button type="button" onClick={validateSignUp}>

          Sign Up

        </s-button>

      </s-form>

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

       <p>Already have an account? </p>

       {/\* <Link to="/login">Log In</Link> \*/}

    </div>

  );

};

export default SignUp;

/\* SignUp.css \*/

body {

    background-image: url('https://source.unsplash.com/1600x900/?waterfront'); /\* Replace with your desired background image URL \*/

    background-size: cover;

    background-position: center;

    background-repeat: no-repeat;

    margin: 0; /\* Remove default margin to cover the entire viewport \*/

  }

  .scontainer {

      background-color: #000404;

      padding: 90px;

      border-radius: 10px;

      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

      max-width: 500px;

      margin-left: 40px; /\*to move the box right \*/

      height: 300px;

      opacity: 80%;

      /\* background-image: url('https://source.unsplash.com/800x600/?nature');  to add bg inside box\*/

      /\* display: flex; Use flex container \*/

      /\* justify-content: flex-end;  \*/

    }

    s-form {

      display: flex;

      flex-direction: column;

    }

    h2 {

      text-align: center;

      margin-bottom: 5px;

      margin-top:-65px

    }

    label {

      margin-bottom: 2px;

    }

    input {

      padding: 8px;

      margin-bottom: 16px;

      background-color: #acaeac;

    }

    s-button {

      padding: 10px;

      background-color: #0946ed;

      color: #fcf9f9;

      border: none;

      border-radius: 10px;

      cursor: pointer;

    }

    s-button:hover {

      background-color: #0c1c67;

    }

    #error-message {

      color: rgb(142, 2, 2);

      margin-top: 10px;

      text-align: center;

    }