**Semester Project Task: Design and Implement Login and Registration Forms with php sessions for your project**

**Objective:**

The objective of this task is to create basic login and registration forms for a web-based application. Students will demonstrate their understanding of HTML, CSS, and form validation techniques.

**Task Description:**

Students will design and implement simple login and registration forms using HTML and CSS. The forms should include basic input fields and validation for username/email and password.

**Login Form:**

Username/email and password fields.

Basic CSS styling for visual appeal.

Simple client-side validation to ensure fields are not empty.

Registration Form:

Username, email, password, and confirm password fields.

Basic CSS styling for visual appeal.

Simple client-side validation to ensure fields are not empty and passwords match.

**Evaluation Criteria:**

**Functionality:**

Login form allows users to input their credentials.

Registration form allows users to input their information and register.

**Validation:**

Forms have basic client-side validation to ensure required fields are not empty.

**Usability:**

Forms have a clear layout and are easy to use.

Error messages are displayed when fields are left empty or passwords do not match.

**Submission:**

Students will submit the HTML and CSS files for both the login and registration forms.

Jsx File:

import React, { useState } from 'react';

import './LoginSignup.css';

import user\_icon from '../Assets/person.png';

import email\_icon from '../Assets/email.png';

import password\_icon from '../Assets/password.png';

const LoginSignup = () => {

    const [action, setAction] = useState("Login");

    const initialValues = { username: "", email: "", password: "", cpassword: "" };

    const [formValues, setFormValues] = useState(initialValues);

    const [isSubmit, setIsSubmit] = useState(false);

    const [errors, setErrors] = useState({});

    const handleChange = (e) => {

        const { name, value } = e.target;

        setFormValues({ ...formValues, [name]: value });

    };

    const handleSubmit = (e) => {

        e.preventDefault();

        const validationErrors = validate(formValues);

        if (Object.keys(validationErrors).length === 0) {

            setIsSubmit(true);

        } else {

            setErrors(validationErrors);

        }

    };

    const validate = (values) => {

        let errors = {};

        if (values.username.trim() === "") {

            errors.username = "Username cannot be empty";

        }

        if (action === "Sign Up") {

            if (values.email.trim() === "") {

                errors.email = "Email cannot be empty";

            } else if (!/\S+@\S+\.\S+/.test(values.email)) {

                errors.email = "Invalid email address";

            }

            if (values.password.trim() === "") {

                errors.password = "Password cannot be empty";

            }

            if (values.cpassword.trim() === "") {

                errors.cpassword = "Password cannot be empty";

            } else if (values.cpassword !== values.password) {

                errors.cpassword = "Passwords do not match";

            }

        } else {

            if (values.password.trim() === "") {

                errors.password = "Password cannot be empty";

            }

        }

        return errors;

    };

    return (

        <div className='container'>

            <div className="header">

                <div className="text">{action}</div>

            </div>

            <form onSubmit={handleSubmit}>

                <div className="inputs">

                    <div className="input">

                        <img src={user\_icon} alt="" />

                        <input onChange={handleChange} type="text" name="username" placeholder='Username' value={formValues.username} />

                        {errors.username && <div className="error">{errors.username}</div>}

                    </div>

                    {action === "Login" ? null :

                        <div className="input">

                            <img src={email\_icon} alt="" />

                            <input onChange={handleChange} type="email" name="email" placeholder='Email' value={formValues.email} />

                            {errors.email && <div className="error">{errors.email}</div>}

                        </div>

                    }

                    <div className="input">

                        <img src={password\_icon} alt="" />

                        <input onChange={handleChange} type="password" name="password" placeholder='Password' value={formValues.password} />

                        {errors.password && <div className="error">{errors.password}</div>}

                    </div>

                    {action === "Login" ? null :

                        <div className="input">

                            <img src={password\_icon} alt="" />

                            <input onChange={handleChange} type="password" name="cpassword" placeholder='Confirm Password' value={formValues.cpassword} />

                            {errors.cpassword && <div className="error">{errors.cpassword}</div>}

                        </div>

                    }

                </div>

                <div className="submit-container">

                    <button type="submit" className="submit">{action === "Login" ? "Log in" : "Sign Up"}</button>

                </div>

                {action === "Login" ?

                    <div className="register" onClick={() => { setAction("Sign Up") }}>New to CMS?<span> Register</span></div> :

                    <div className="register" onClick={() => setAction("Login")}>Already have an account?<span>Login</span></div>

                }

            </form>

        </div>

    );

};

export default LoginSignup;

CSS File:

.container{

    display:flex;

    flex-direction: column;

    margin:auto;

    margin-top: 50px;

    width:450px;

    background:white;

    padding-bottom:30px;

}

.header{

    display: flex;

    flex-direction: column;

    align-items: center;

    gap: 9px;

    width:100%;

    margin-top:30px;

}

.text{

    color:#0066ff;

    font-size: 35px;

    font-weight: 700;

}

 .inputs{

    margin-top:55px;

    display:flex;

    flex-direction: column;

    gap:25px;

 }

 .input{

    display:flex;

    align-items: center;

    margin:auto;

    width:350px;

    height:70px;

    background:lightgray;

    border-radius: 6px;

 }

 .input img{

    margin: 0px 30px;

 }

 .input input{

    height:50px;

    width:400px;

    background: transparent;

    border: none;

    outline: none;

    color:black;

    font-size:15px;

 }

 .submit-container{

    display: flex;

    justify-content: center;

    gap:9px;

    margin:20px auto;

 }

 .submit{

    display: flex;

    justify-content: center;

    align-items: center;

    width: 150px;

    height:59px;

    color:white;

    background-color: #0066ff;

    border-radius: 50px;

    font-size: 15px;

    font-weight: 700;

    cursor: pointer;

 }

 .register{

    display: flex;

    justify-content: center;

 }

 .register span{

    display: flex;

    justify-content: center;

    color:#0066ff;

    cursor:pointer;

 }

 .error{

    color:red;

 }

Screenshots:

A screenshot of a computer

Description automatically generated

A screenshot of a login form

Description automatically generated

A screenshot of a login form

Description automatically generated