import React, { useState } from "react";

**interface** FormData {

username: string;

password: string;

gender: string;

country: string;

agreement: boolean;

}

**interface** ErrorMsg {

username: string;

password: string;

gender: string;

country: string;

agreement: string;

}

const App = () => {

const initialFormData: **FormData** = {

username: '',

password: '',

gender: '',

country: '',

agreement: false,

};

const initialErrorData: **ErrorMsg** = {

username: '',

password: '',

gender: '',

country: '',

agreement: '',

};

const [formData, setFormData] = useState**<FormData>(initialFormData)**;

const [errorMsg, setErrorMsg] = useState**<ErrorMsg>(initialErrorData)**;

const [formSubmitted, setFormSubmitted] = useState(false);

const **handleInputChange** = (event: React.ChangeEvent**<HTMLInputElement | HTMLSelectElement>**) => {

const { name, value, type, checked } = event.target;

**setFormData**((prevFormData) => {

if (type === "checkbox") {

return {

...**prevFormData**,

[name]: checked,

};

} else {

return {

...**prevFormData**,

[name]: value,

};

}

});

if (formSubmitted) {

setErrorMsg({

...errorMsg,

[name]: '',

});

}

};

const **handleLogin** = (event: React.FormEvent<HTMLFormElement>) => {

event.preventDefault();

// Reset all error messages

**setErrorMsg**({

username: '',

password: '',

gender: '',

country: '',

agreement: '',

});

// Check for empty fields and update error messages

**(Object.keys(formData) as (keyof FormData)[]).forEach((item) => {**

**if (!formData[item]) {**

**setErrorMsg((prevErrorMsg) => ({**

**...prevErrorMsg,**

**[item]: `${item.charAt(0).toUpperCase() + item.slice(1)} is required`,**

**}));**

**}**

**});**

// If there are no errors, proceed with the login

if **(!Object.values(errorMsg).some((msg) => msg !== '')**) {

console.log('Login successful!');

// Perform login logic here

// For example, redirect to a dashboard or show a success message to the user

}

// Set the formSubmitted flag to true after the form is submitted

**setFormSubmitted**(true);

};

return (

<div>

<h1>Login</h1>

<form onSubmit={handleLogin}>

<div>

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

**<input type="text" id="username" name="username" value={formData.username} onChange={handleInputChange}/>**

<p className="error">{formSubmitted && errorMsg.username}</p>

</div>

<div>

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

**<input type="password" id="password" name="password" value={formData.password} onChange={handleInputChange}/>**

<p className="error">{formSubmitted && errorMsg.password}</p>

</div>

<div>

<label>Gender</label>

<div>

**<input type="radio" id="male" name="gender" value="male" checked={formData.gender === "male"} onChange={handleInputChange}/>**

<label htmlFor="male">Male</label>

**<input type="radio" id="female" name="gender" value="female" checked={formData.gender === "female"} onChange={handleInputChange}/>**

<label htmlFor="female">Female</label>

</div>

<p className="error">{formSubmitted && errorMsg.gender}</p>

</div>

<div>

<label htmlFor="country">Country</label>

**<select id="country" name="country" value={formData.country} onChange={handleInputChange}>**

**<option value="">Select Country</option>**

**<option value="usa">USA</option>**

**<option value="canada">Canada</option>**

**{/\* Add more options as needed \*/}**

**</select>**

<p className="error">{formSubmitted && errorMsg.country}</p>

</div>

<div>

**<input type="checkbox" id="agreement" name="agreement" checked={formData.agreement} onChange={handleInputChange}/>**

<label htmlFor="agreement">I agree to the terms and conditions</label>

<p className="error">{formSubmitted && errorMsg.agreement}</p>

</div>

<div>

<button type="submit">Login</button>

</div>

</form>

</div>

);

};

export default App;