import React, { useState } from "react";

import { Form, Button, Input, DatePicker } from "antd";

const layout = {

labelCol: { span: 8 },

wrapperCol: { span: 16 },

};

const tailLayout = {

wrapperCol: { offset: 8, span: 16 },

};

const UserForm = () => {

const [formData, setFormData] = useState({

name: "",

dateOfBirth: null,

email: "",

phoneNumber: "",

});

const [formErrors, setFormErrors] = useState({

name: "",

dateOfBirth: "",

email: "",

phoneNumber: "",

});

const handleChange = (e) => {

setFormData({ ...formData, [e.target.name]: e.target.value });

validateField(e.target.name, e.target.value);

};

const handleDateChange = (date, dateString) => {

setFormData({ ...formData, dateOfBirth: dateString });

validateField("dateOfBirth", dateString);

};

const validateField = (fieldName, value) => {

switch (fieldName) {

case "name":

setFormErrors({ ...formErrors, name: value ? "" : "Name is required" });

break;

case "dateOfBirth":

const today = new Date();

const birthDate = new Date(value);

const age = today.getFullYear() - birthDate.getFullYear();

const m = today.getMonth() - birthDate.getMonth();

if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {

age--;

}

setFormErrors({ ...formErrors, dateOfBirth: age >= 18 ? "" : "Age must be at least 18" });

break;

case "email":

setFormErrors({ ...formErrors, email: value.match(/^([\w.%+-]+)@([\w-]+\.)+([\w]{2,})$/i) ? "" : "Invalid email address" });

break;

case "phoneNumber":

setFormErrors({ ...formErrors, phoneNumber: value.match(/^\d{10}$/) ? "" : "Invalid phone number" });

break;

default:

break;

}

};

const isFormValid = () => {

return Object.values(formErrors).every((error) => error === "");

};

return (

<Form

{...layout}

name="basic"

initialValues={formData}

onFinish={(values) => {

console.log("Form submitted: ", values);

}}

>

<Form.Item

label="Name"

name="name"

rules={[{ required: true, message: "Please input your name!" }]}

validateStatus={formErrors.name ? "error" : ""}

help={formErrors.name}

>

<Input name="name" onChange={handleChange} />

</Form.Item>

<Form.Item

label="Date of Birth"

name="dateOfBirth"

rules={[{ required: true, message: "Please select your date of birth!" }]}

validateStatus={formErrors.dateOfBirth ? "error" : ""}

help={formErrors.dateOfBirth}

>

<DatePicker onChange={handleDateChange} />

</Form.Item>

<Form.Item

label="Email"

name="email"

rules={[{ required: true, message: "Please input your email!" }]}

validateStatus={formErrors.email ? "error" : ""}

help={formErrors.email}

>

<Input name="email" onChange={handleChange} />

</Form.Item>

<Form.Item

label="Phone Number"

name="phoneNumber"

rules={[{ required: true, message: "Please input your phone number!" }]}

validateStatus={formErrors.phoneNumber ? "error" : ""}

help={formErrors.phoneNumber}

>

<Input name="phoneNumber" onChange={handleChange} />

</Form.Item>

<Form.Item {...tailLayout}>

<Button type="primary" htmlType="submit" disabled={!isFormValid()}>

Submit

</Button>

</Form.Item>

</Form>

);