**1. Configurar EmailJS**

1. **Crear una Cuenta en EmailJS:** Regístrate en [EmailJS](https://www.emailjs.com/). Es un servicio que permite enviar correos electrónicos desde el frontend.
2. **Configurar un Servicio de Email:** Una vez registrado, ve a la sección de "Email Services" y configura el servicio de correo que utilizarás (por ejemplo, Gmail, Outlook, etc.).
3. **Obtener las Credenciales de EmailJS:**
   * **ID del Servicio de Email**: Obtén el ID del servicio que configuraste.
   * **ID del Plantilla**: Crea una plantilla de correo en la sección de "Email Templates" y obtén el ID de la plantilla.
   * **ID del Usuario**: Encuentra tu ID de usuario en la configuración de tu cuenta.

**2. Instalar EmailJS en el Proyecto**

Instala el paquete emailjs-com en tu proyecto React:

bash

Copiar código

npm install emailjs-com

**3. Crear el Formulario de Contacto**

Actualiza Contact.js para incluir un formulario de contacto que envíe un correo electrónico cuando se envíe el formulario.

**Contact.js**

javascript

Copiar código

// src/pages/Contact.js

import React, { useState } from 'react';

import { Form, Button, Alert } from 'react-bootstrap';

import emailjs from 'emailjs-com';

const Contact = () => {

const [formData, setFormData] = useState({

name: '',

email: '',

message: ''

});

const [sent, setSent] = useState(false);

const [error, setError] = useState('');

const handleChange = (e) => {

setFormData({

...formData,

[e.target.name]: e.target.value

});

};

const handleSubmit = (e) => {

e.preventDefault();

emailjs.sendForm('YOUR\_SERVICE\_ID', 'YOUR\_TEMPLATE\_ID', e.target, 'YOUR\_USER\_ID')

.then((result) => {

setSent(true);

setFormData({

name: '',

email: '',

message: ''

});

}, (error) => {

setError('There was an error sending your message. Please try again.');

});

};

return (

<div className="container mt-4">

<h1 className="page-title">Contact Us</h1>

{sent && <Alert variant="success">Your message has been sent successfully!</Alert>}

{error && <Alert variant="danger">{error}</Alert>}

<Form onSubmit={handleSubmit}>

<Form.Group controlId="formName">

<Form.Label>Name</Form.Label>

<Form.Control

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</Form.Group>

<Form.Group controlId="formEmail">

<Form.Label>Email</Form.Label>

<Form.Control

type="email"

name="email"

value={formData.email}

onChange={handleChange}

required

/>

</Form.Group>

<Form.Group controlId="formMessage">

<Form.Label>Message</Form.Label>

<Form.Control

as="textarea"

rows={3}

name="message"

value={formData.message}

onChange={handleChange}

required

/>

</Form.Group>

<Button variant="primary" type="submit">

Send

</Button>

</Form>

</div>

);

};

export default Contact;

**Actualizar Variables en emailjs.sendForm**

Reemplaza YOUR\_SERVICE\_ID, YOUR\_TEMPLATE\_ID, y YOUR\_USER\_ID con los valores correspondientes obtenidos de EmailJS.

**4. Ejecutar la Aplicación**

Después de hacer estos cambios, ejecuta la aplicación nuevamente:

bash

Copiar código

npm start

Ahora, tu página de "Contact" debería mostrar un formulario de contacto que envía un correo electrónico con la información del cliente a contact@xyncro.com cuando se envía el formulario.

4o mini