[ChatGPT (openai.com)](https://chat.openai.com/c/b90b1868-7c00-4da3-b1f7-c2cd516a7471)

Para implementar un formulario de contacto en tu sitio web que permita a los usuarios enviarte correos electrónicos directamente, necesitarás seguir una serie de pasos tanto en el lado del cliente (HTML) como en el servidor (Node.js con Express y Nodemailer). Aquí tienes una guía paso a paso considerando el cambio solicitado:

### Paso 1: Preparación del Entorno

1. **Instala Node.js**: Si aún no lo tienes instalado, descárgalo e instálalo desde [la página oficial de Node.js](https://nodejs.org/).
2. **Crea un Directorio para tu Proyecto**: En tu sistema, crea un nuevo directorio para tu proyecto y navega a él en la terminal o línea de comandos.
3. **Inicializa un Nuevo Proyecto de Node.js**:
   * Ejecuta **npm init -y** en la terminal para crear un nuevo proyecto de Node.js. Esto generará un archivo **package.json** con la configuración predeterminada.

### Paso 2: Instalar Dependencias

Dentro del directorio de tu proyecto, instala las siguientes dependencias ejecutando los comandos proporcionados:

1. **Express**: Un framework de Node.js para construir aplicaciones web.
   * **npm install express**
2. **Nodemailer**: Una librería para enviar correos electrónicos fácilmente.
   * **npm install nodemailer**
3. **Body-Parser**: Middleware para analizar cuerpos de solicitud entrantes en Express.
   * **npm install body-parser**

### Paso 3: Crear el Servidor de Express

Crea un archivo, por ejemplo, **server.js**, en el directorio de tu proyecto y añade el siguiente código:

javascriptCopy code

const express = require('express'); const nodemailer = require('nodemailer'); const bodyParser = require('body-parser'); const app = express(); const port = 3000; app.use(bodyParser.urlencoded({ extended: true })); app.post('/enviarCorreo', (req, res) => { let transporter = nodemailer.createTransport({ service: 'gmail', auth: { user: 'tuCorreo@gmail.com', pass: 'tuClaveDeAplicacion' } }); let mailOptions = { from: req.body.correo, to: 'tuCorreo@gmail.com', subject: `Mensaje de ${req.body.nombre}`, text: req.body.mensaje }; transporter.sendMail(mailOptions, (error, info) => { if (error) { console.log(error); res.send('Error al enviar el mensaje'); } else { console.log('Mensaje enviado: %s', info.messageId); res.send('Mensaje enviado con éxito'); } }); }); app.listen(port, () => { console.log(`Servidor ejecutándose en http://localhost:${port}`); });

### Paso 4: Crear el Formulario HTML

Integra el siguiente formulario en tu archivo HTML en la sección correspondiente:

htmlCopy code

<section class="page-section" id="contact"> <div class="container"> <!-- Aquí iría el encabezado de la sección y cualquier otro elemento que desees incluir --> <form action="http://localhost:3000/enviarCorreo" method="POST"> <div class="form-group"> <input class="form-control" name="nombre" type="text" placeholder="Tu nombre..." required /> </div> <div class="form-group"> <input class="form-control" name="correo" type="email" placeholder="Tu correo electrónico..." required /> </div> <div class="form-group"> <textarea class="form-control" name="mensaje" placeholder="Tu mensaje..." required></textarea> </div> <button type="submit" class="btn btn-primary">Enviar</button> </form> </div> </section>

### Paso 5: Ejecutar el Servidor

En la terminal o línea de comandos, navega al directorio de tu proyecto y ejecuta el servidor con el comando:

bashCopy code

node server.js

### Paso 6: Prueba el Formulario

Abre tu navegador y ve a la página donde está alojado tu formulario. Completa y envía el formulario para ver si recibes el correo electrónico.

### Notas Importantes:

* Asegúrate de reemplazar **tuCorreo@gmail.com** y **tuClaveDeAplicacion** con tu dirección de correo electrónico de Gmail y la clave de aplicación que obtienes de tu cuenta de Google.
* Este ejemplo asume que estás ejecutando el servidor localmente. Si despliegas tu aplicación, asegúrate de actualizar la acción del

Principio del formulario

Servidor de aplicación

rubiwankenobi@gmail.com

zifk iwdl reux sokq