Example: Send an Email Using Nodemailer

// Step 1: Import the Nodemailer package const nodemailer = require('nodemailer');

// Step 2: Create a transporter object using SMTP configuration const transporter = nodemailer.createTransport({ service: 'gmail', // or use 'smtp.your-email-provider.com' auth: { user: 'your_email@gmail.com', // your email address pass: 'youremailpasswordorappspecificpassword' // use app password for Gmail } });

// Step 3: Define the email details (sender, recipient, subject, message) const mailOptions = { from: 'your_email@gmail.com', to: 'recipient@example.com', subject: 'Test Email from Node.js', text: 'Hello! This email was sent using Nodemailer in Node.js.' };

// Step 4: Send the email const sendEmail = async () => { try { const info = await transporter.sendMail(mailOptions); console.log('Email sent successfully:', info.messageId); } catch (error) { console.error('Error sending email:', error); } };

// Step 5: Run the function sendEmail();

Setup Requirements

Before running:

1. Install Nodemailer:

   ```
   npm install nodemailer
   ```

2. Enable “Less Secure Apps” or use an App Password if you’re using Gmail with 2FA.
3. Replace the placeholders (`your_email@gmail.com` , etc.) with real values.

Notes

- For production, always store credentials in environment variables (`process.env.EMAILUSER` , `process.env.EMAILPASS`).
- You can also use external SMTP servers (like SendGrid, Mailgun, or AWS SES) by changing the service and SMTP configuration.