

```

const express =
  require('express');
const multer =
  require('multer');
const nodemailer =
  require('nodemailer');
const fs = require('fs');
const path =
  require('path');
require('dotenv').config();

const app = express();

// Configure multer for file
// uploads
const upload = multer({
  dest: 'uploads/',
  limits: { fileSize: 10 *
    1024 * 1024 }, // 10 MB
  limit,
  fileFilter: (req, file, cb) =>
  {
    const allowedTypes =
    ['application/pdf',
    'image/jpeg', 'image/png',
    'video/mp4'];
    if
    (!allowedTypes.includes(file.
    mimetype)) {
      return cb(new
      Error('Only PDFs, JPEGs,
      PNGs, and MP4s are
      allowed.'));
    }
    cb(null, true);
  }
});

app.post('/upload',
  upload.array('files'), async
  (req, res) => {
    try {
      const email =
      req.body.email ||
      'Anonymous';
      const files = req.files;

      if (!files || files.length
      === 0) {
        return
        res.status(400).json({
        message: 'No files
        uploaded.' });
      }

      // Configure
      // Nodemailer
      const transporter =
      nodemailer.createTransport({
        service: 'Gmail',
        auth: {
          user:
            process.env.EMAIL_USER,
          pass:
            process.env.EMAIL_PASS
        }
      });

      // Email attachments
      const attachments =
      files.map(file => ({
        filename:
          file.originalname,
        path: file.path
      }));

      // Send email
      await
      transporter.sendMail({
        from:
          process.env.EMAIL_USER,
        to:
          'omaralnahas2010@gmail
          .com',
        subject: 'New File
        Submission',
        text: `New files
        uploaded by: ${email}`,
        attachments
      });

      // Cleanup files
      files.forEach(file =>
      fs.unlinkSync(file.path));

      res.status(200).json({
        message: 'Files uploaded
        and email sent
        successfully!' });
    } catch (err) {
      console.error(err.message);
      res.status(500).json({
        message: 'Error
        processing your request.'
      });
    }
  });

// Start server
const PORT =
  process.env.PORT || 3000;
app.listen(PORT, () =>
  console.log(`Server
  running on port
  ${PORT}`));

```