A screen shot of a file upload

Description automatically generated

const http = require('http');

const formidable = require('formidable');

const fs = require('fs');

const path = require('path');

http.createServer((req, res) => {

  if (req.url === '/fileupload' && req.method.toLowerCase() === 'post') {

    const form = new formidable.IncomingForm();

    form.parse(req, (err, fields, files) => {

      if (err) {

        res.writeHead(500, {'Content-Type': 'text/plain'});

        res.end('Error occurred during file upload');

        return;

      }

      const file = files.filetoupload;

      // Ensure the uploaded file is an image

      const allowedTypes = ['image/jpeg', 'image/png', 'image/gif'];

      if (!allowedTypes.includes(file.mimetype)) {

        res.writeHead(400, {'Content-Type': 'text/plain'});

        res.end('Only image files are allowed!');

        return;

      }

      const oldpath = file.filepath;

      const newpath = path.join(\_\_dirname, 'upload', file.originalFilename);

      fs.rename(oldpath, newpath, (err) => {

        if (err) {

          res.writeHead(500, {'Content-Type': 'text/plain'});

          res.end('Error occurred while moving the file');

          return;

        }

        res.writeHead(200, {'Content-Type': 'text/plain'});

        res.end('File uploaded and moved!');

      });

    });

  } else {

    res.writeHead(200, {'Content-Type': 'text/html'});

    res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');

    res.write('<input type="file" name="filetoupload"><br>');

    res.write('<input type="submit">');

    res.write('</form>');

    res.end();

  }

}).listen(3000, () => {

  console.log('Server is listening on port 3000');

});

Screenshots  
A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with a black background

Description automatically generated