

```
const fs = require("fs");
const path = require("path");
const axios = require("axios");
const FormData = require("form-data");
const watch = require("node-watch");

const LOCAL_FOLDER = "D:/upload";
const UPLOAD_URL = "https://mvrezab.com/upload.php";
const FILE_STABLE_DELAY = 2000;

// Keep track of uploaded files
const uploadedFiles = new Set();

console.log("Watching folder:", LOCAL_FOLDER);

watch(LOCAL_FOLDER, { recursive: true }, (evt, filePath) => {
  if (filePath.endsWith(".crdownload")) return;
  if (evt !== "update") return;

  // If already uploaded, skip
  if (uploadedFiles.has(filePath)) return;

  // Wait for file to be stable
  setTimeout(async () => {
    if (!fs.existsSync(filePath)) return;

    console.log("New file detected:", filePath);

    const form = new FormData();
    form.append("file", fs.createReadStream(filePath));

    try {
      const response = await axios.post(UPLOAD_URL, form,
      {
        headers: form.getHeaders(),
      });

      console.log("Uploaded successfully:", response.data);
      uploadedFiles.add(filePath); // mark as uploaded
    } catch (err) {

```

```

        console.error("Upload failed:", err.message);
    }
}, FILE_STABLE_DELAY);
}); do you have any professional app or someting for
similar this

```

Yeah—what you've built is basically a **custom auto-sync / auto-upload agent**, and there **are** very solid **professional tools** that already do this (often more reliably, with retries, logs, deletes, versioning, etc.).

I'll break it down by **use case**, so you can pick what fits you best 

◆ Best Professional Tools (No Coding Needed)

Rclone (🔥 Highly Recommended)

Closest to your script, but enterprise-grade

Why it's great

- Watches folders
- Auto upload / sync
- Supports delete sync (`--delete`)
- Retry, checksum, resume
- Works with FTP, SFTP, WebDAV, cloud, custom servers

Example (same behavior as your script):

bash

```
rclone sync D:/upload remote:uploads --progress --delete-during
```

 Copy code

 This alone replaces 90% of custom Node scripts

 Used by devs, sysadmins, startups

Syncthing

Real-time two-way sync

- Folder watch 
- Delete sync 

- Encrypted ✓
- No server needed (peer-to-peer)
- GUI dashboard

⚠ Best when both machines are yours

✓ GoodSync

Commercial / enterprise solution

- Folder monitoring
- Upload + delete sync
- FTP / SFTP / WebDAV
- Scheduling + logs
- GUI-based

💰 Paid, but very stable

✓ WinSCP (Automation Mode)

If your server supports SFTP / FTP:

- Scriptable
- Folder sync
- Delete propagation
- Very popular in Windows environments

◆ Developer / Cloud-Level Solutions

■ AWS S3 + AWS DataSync

Enterprise-grade auto upload + delete + versioning

(Overkill unless you're scaling big)

◆ When Your Node.js Script Still Makes Sense

Your script is good if:

- You need **custom PHP API logic**
- You want metadata / DB logging

- You want authentication tokens
- You want full control

But for **production**, you'd still need to add:

- Retry queue
- Hash/checksum verification
- Upload resume
- Delete sync logic
- Crash recovery
- Logging

That's exactly what tools like **Rclone** already solved.

🔥 My Honest Recommendation (as a dev)

If your goal is:

Just auto upload + delete sync

→ Use Rclone

Two-way live sync

→ Syncthing

Enterprise / client delivery

→ GoodSync

Learning / custom backend

→ Keep your Node.js watcher

If you want, I can:

- Convert your setup **fully** to Rclone
- Add **delete.php** + Node delete sync
- Make your Node app **production-ready**
- Turn this into a **Windows background service**

Just tell me what direction you want to go 



in bangla

ঠিক আছে 

তোমার এই auto file upload + folder watch কাজটার জন্য বাজারে আসলেই কিছু professional, stable এবং trusted software আছে। নিচে সহজ করে বাংলায় বুঝিয়ে দিচ্ছি।

Professional Software (Coding ছাড়াই)

 Rclone (সবচেয়ে Best)

 তোমার Node.js script-এর professional version ভাবতে পারে

কি কি পারে

- Folder watch করে
- Automatic upload করে
- Local folder থেকে file delete হলে server থেকেও delete করে
- Retry, resume, checksum সব আছে
- FTP / SFTP / WebDAV / Cloud সব support করে

Example (তোমার কোডের মতো কাজ):

bash

```
rclone sync D:/upload remote:uploads --delete-during --progress
```

 Copy code

 Production server গুলোতে সবচেয়ে বেশি use হয়

 Dev + SysAdmin দুজনই ব্যবহার করে

Syncthing

Real-time দুই পাশে sync

- File add/update/delete → সাথে সাথে sync
- Encrypted
- GUI dashboard আছে
- Coding দরকার নাই

 Server আর PC দুটোই তোমার হলে best

GoodSync (Paid)

- Folder monitor
- Auto upload + auto delete
- FTP / SFTP support
- Log, schedule, GUI

 Company / office environment এ বেশি use হয়

WinSCP (Automation)

যদি তোমার server এ FTP/SFTP থাকে:

- Script দিয়ে auto sync করা যায়
- Stable & popular
- Windows user দের জন্য ভালো

তাহলে তোমার Node.js script কেন দরকার?

তোমার script ভালো হবে যদি:

- Custom upload.php / delete.php ব্যবহার করতে চাও
- Database update দরকার
- Token / Auth দরকার
- Server-side logic control করতে চাও

কিন্তু production এর জন্য তোমার script এ আরো লাগবে:

- Retry system
- Upload queue
- Hash/checksum verify
- Delete sync
- Error log
- Crash recovery

এই সব কিছু Rclone আগেই solve করে রেখেছে।

আমার honest recommendation

শুধু upload + delete sync দরকার?

→ Rclone ব্যবহার করো (BEST)

✓ Real-time two-way sync দরকার?

→ Syncthing

✓ Client / office কাজ?

→ GoodSync

✓ শেখার জন্য বা custom backend?

→ তোমার Node.js script ঠিক আছে

চাও তো আমি:

- তোমার জন্য Rclone full setup করে দিতে পারি
- Node.js + delete.php complete code দিতে পারি
- এটাকে Windows background service বানাতে পারি

বললেই হলো 

     ...

The system cannot find the path specified.

node:internal/modules/cjs/loader:1423

 throw err;

 ^

Error: Cannot find module 'p-queue'

Require stack:

 - D:\auto-upload\upload-watcher.js

 at Module._resolveFilename

 (node:internal/modules/cjs/loader:1420:15)

 at defaultResolveImpl

 (node:internal/modules/cjs/loader:1058:19)

 at resolveForCJSWithHooks

 (node:internal/modules/cjs/loader:1063:22)

 at Module._load

 (node:internal/modules/cjs/loader:1226:37)

 at TracingChannel.traceSync

 (node:diagnostics_channel:328:14)

```

at wrapModuleLoad
(node:internal/modules/cjs/loader:244:24)
  at Module.require
(node:internal/modules/cjs/loader:1503:12)
  at require (node:internal/modules/helpers:152:16)
  at Object.<anonymous> (D:\auto-upload\upload-
watcher.js:5:16)
  at Module._compile
(node:internal/modules/cjs/loader:1760:14) {
  code: 'MODULE_NOT_FOUND',
  requireStack: [ 'D:\auto-upload\upload-watcher.js' ]
}

```

Node.js v24.11.0
Press any key to continue . . .

ঠিক আছে, আমি বাংলা ভাষায় ব্যাখ্যা করছি। 😊

আপনি যে এরটি পাচেন সেটা মূলত বলছে যে Node.js আপনার প্রজেক্টে p-queue মডিউলটি খুঁজে পাচ্ছে না।

এর মানে:

javascript

Error: Cannot find module 'p-queue'

 Copy code

সমাধান:

1. প্রথমে নিশ্চিত করুন যে আপনি আপনার প্রজেক্টের ফোল্ডারে আছেন। আপনার টার্মিনালে লিখুন:

bash

 Copy code

cd D:\auto-upload

2. এরপর p-queue মডিউলটি ইনস্টল করুন:

bash

 Copy code

npm install p-queue

3. ইনস্টলেশনের পর আবার আপনার স্ক্রিপ্ট চালান:

bash

 Copy code

```
node upload-watcher.js
```

একটু প্রফেশনাল হ্যান্ডেলিং:

p-queue এর মাধ্যমে আপনি একসাথে অনেক ফাইল আপলোড করলে কনকারেন্সি কন্ট্রোল করতে পারবেন, অর্থাৎ একসাথে অনেকগুলো ফাইল সার্ভারে পাঠানো হলে সার্ভার ক্র্যাশ বা ব্যন্ডউইথ সমস্যা হবে না।

যদি আপনি চান, আমি আপনাকে আপনার কোড p-queue দিয়ে প্রফেশনাল, রিয়েল-টাইম ফাইল আপলোড ও ডিলিট হ্যান্ডলার সহ পূর্ণ সংস্করণ বানিয়ে দিতে পারি। এতে আপনার ক্রিপ্ট আরো স্টেবল ও প্রফেশনাল হবে।

আপনি কি আমি সেটা বানিয়ে দেই?

      ...