

# I ran out of disk space, so I shipped a **paid app** to curate a lifetime of photos.

KeepRoll started as my own problem and became a product. A local-first Mac app that curates media across iCloud, external drives, and local folders, surfaces the best shot of a burst, and syncs your keepers back to your phone.

Role: **Solo, built with AI**

Live product: **keeproll.app**

**239** commits in ~17 days

Python, zero-build



KeepRoll, live at [keeproll.app](https://keeproll.app). Swipe to keep or trash across every place your photos live.

---

## The problem (it was mine)

I ran out of space on a 2TB laptop. I did a storage audit with Claude, which flagged a mountain of video and multimedia and suggested moving it to the cloud. My reaction: that is like *moving a house I do not want into another house*. The real problem was not storage. It was that a decade of photos and video was scattered across my phone, my desktop, and several drives, thick with duplicates and forgotten moments, and I had never actually *curated* any of it.

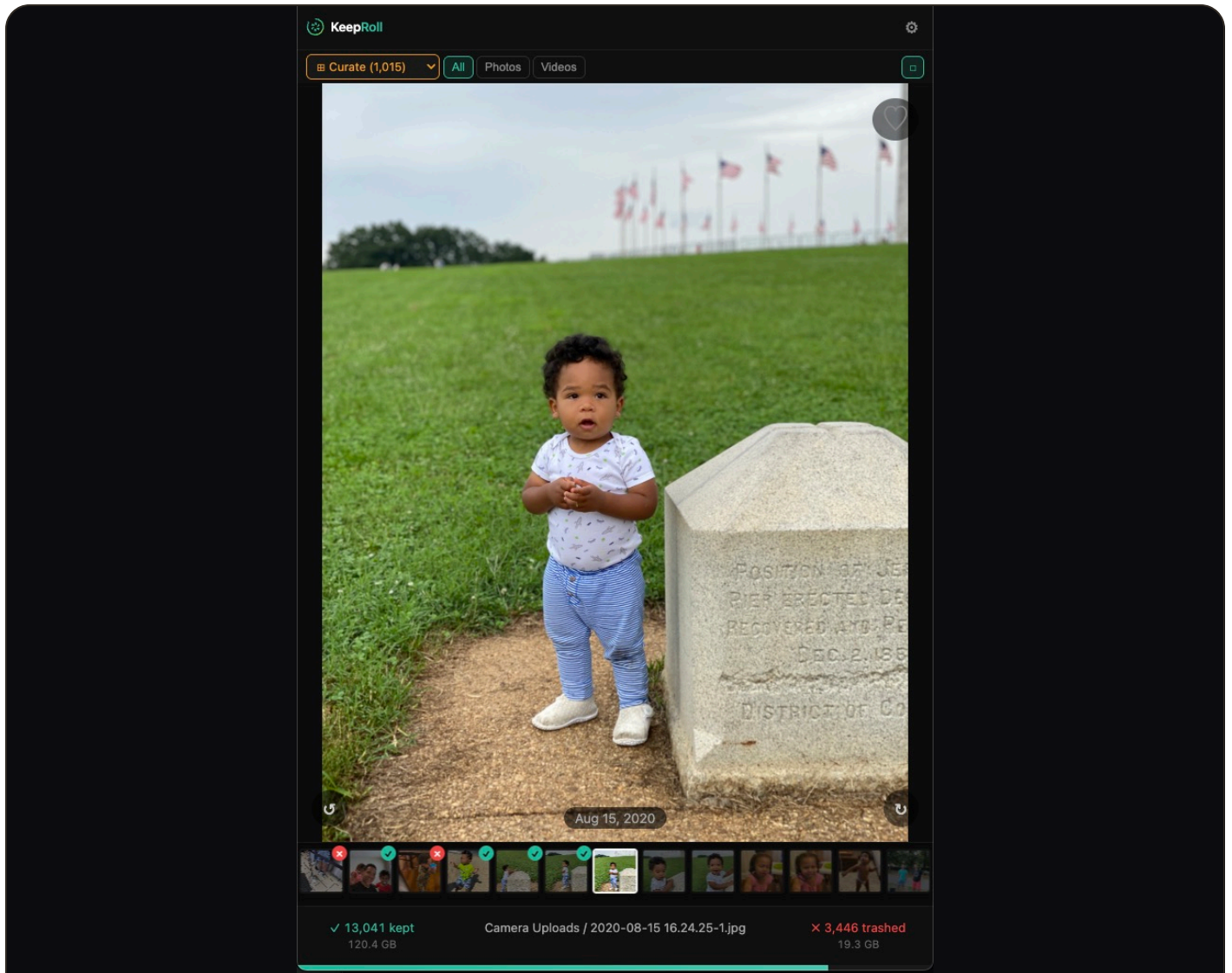
Apple Photos only manages what lives inside Apple Photos. Nothing I could find handled media **across** external drives, local folders, and iCloud at once, and then let me make fast keep-or-lose decisions without dread. So I built it.

---

## What I built

KeepRoll is a local-first photo-curation app for the Mac. You point it at everywhere your media lives and it gives you one fast, calm place to curate it:

- **Curate across every source.** iCloud, external drives, and local folders in one library, not just what is trapped inside Photos.
- **Swipe keep or trash.** A fast, tender flow for making thousands of small decisions without it feeling like a chore.
- **Surface the best of a burst.** Beyond exact duplicates, perceptual-hash clustering groups near-identical shots so you pick the keeper and drop the rest.
- **Share the work.** Share links let someone else help curate a shared library (my wife curated ours).
- **Sync keepers back.** Your curated set syncs back to the iPhone, so the cleanup actually reaches the device you use.



The curation view: one decision at a time, with running kept and trashed totals across a 120 GB library.

---

## Try it yourself

This is the actual KeepRoll interface, running live below on a sample library. On a computer, use the **arrow keys** or the on-screen buttons to keep, trash, or archive; on a phone, **swipe**. It is the same curation flow the real app uses.

A sandboxed, client-side demo. The real app runs on your own Mac across iCloud, external drives, and local folders.

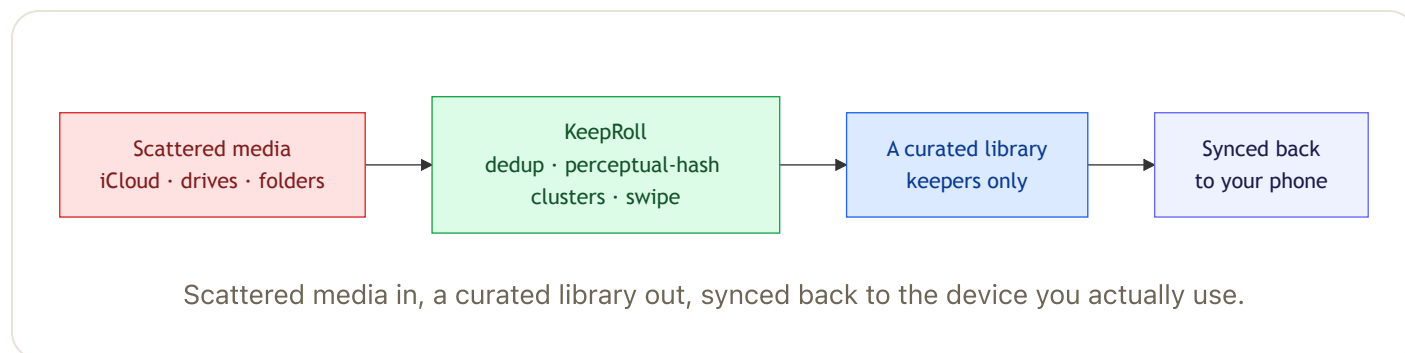
---

## How it is built

The engineering choice I am proudest of is restraint. KeepRoll is deliberately **lightweight**: a Python standard-library HTTP server with an embedded HTML/CSS/JS front end, **no web framework and**

**no build step**, SQLite for state, and **perceptual hashing rather than a machine-learning model** for near-duplicate detection. The core is a 14-mixin `Sorter` architecture that keeps the logic composable.

I built it solo, with Claude Code, in roughly seventeen days across 239 commits. That velocity is the point: a real, shippable, paid product from a standing start in under three weeks, because the tooling let me move at the speed of the idea.



## Why it matters

This is the clearest example of how I work: take a real, felt problem, frame it sharply, and ship a product that resolves it, fast. Most people sit with a “what if I could...” and never build it. I turn that into a live, paid system in weeks. KeepRoll also shows range, it is a polished **consumer product**, not an internal tool: design, packaging, a marketing site, and a checkout, all the way to a paying customer. That instinct, problem to shipped product, is exactly what I bring to a team that has more good ideas than it has time to build.

## By the numbers

**239**

Commits, solo, built with AI

**~17** days

From personal problem to shipped product

**Live**

Paid product at [keeproll.app](https://keeproll.app)

**3**

Sources unified: iCloud, drives, folders

**0**

Build steps, frameworks, or runtime ML

**14**

Mixin Sorter architecture, composable by design

*A personal product, live at [keeproll.app](https://keeproll.app). It is the lead example of how I turn a “what if I could” into a shipped system. Happy to walk through it in a conversation.*

---

Ben Young · builder of KeepRoll

**[benyoung.ai](https://benyoung.ai) · [o.benjamin.young@gmail.com](mailto:o.benjamin.young@gmail.com)**