

ToonTok

By Austen Tucker

Why I built it

A one-off prompt can make an attractive picture while losing the character. I wanted a product where references, approved canon, style choices, and new artwork remain distinct—and where the person using the studio controls what becomes authoritative.

ToonTok turns that problem into a repeatable workflow. Its public showroom explains the product, while invitation-based accounts protect access to private characters, source images, and generation tools.

Build snapshot

- Product surface: Public showroom plus private Vault, Light Table, guides, and transform flows
- Character system: Reusable profiles, references, approved model sheets, and versioned canon
- AI boundary: Server-side image provider, durable jobs, credits, and cost confirmation
- Access: High-contrast interface, large controls, keyboard paths, and 200% zoom support

What I did

I modeled character identity separately from output

Character profiles, reference images, guides, approved model sheets, and generated work have different jobs. ToonTok stores those distinctions instead of flattening everything into one prompt or gallery. A new image can be useful without silently becoming canon.

I made privacy part of the data model

Private assets are owner-scoped and stored behind authenticated access. Transform workflows record consent receipts and source hashes. The provider key stays on the server. These are product boundaries, not promises left to interface copy.

I built review and repair into the workflow

The Light Table gives people a place to inspect work, compare references, accept an image, or request a localized repair. Repair lineage remains attached to the work so a correction does not erase how the result was made.

I made cost visible before the expensive action

Generation, editing, and repair can call paid AI services. ToonTok tracks credits and carries an explicit cost-confirmation field through chargeable operations. The useful lesson was not merely to show a price: confirmation has to be part of the server-validated command.

What I learned

Creative AI needs an authority model

The most important question is not whether a model can generate another image. It is who can approve references, change canon, replace a model sheet, or authorize a repair. Explicit roles and version history make those decisions inspectable.

Privacy changes architecture

“Private by default” affects storage, routes, ownership checks, logs, and what can appear in a public demo. It cannot be added at the end as a hidden gallery setting.

Human acceptance is part of the system

A completed provider request is not an approved creative result. ToonTok separates generation from review and canonization so the human decision remains visible.

Accessibility must survive the real workflow

Large controls and contrast matter, but so do focus order, readable status language, resumable intake, and alternatives to image detail. The product has to remain usable while choosing references, comparing work, and confirming a consequential action.

Current best

ToonTok's strongest product idea is that character continuity is a managed creative asset. The system gives references, consent, canon, cost, generation, and human approval separate places in one end-to-end workflow.

Limits and open questions

- Account creation is invitation-based; the public showroom does not expose private studio data.
- There are no public adoption, reliability, or business-impact numbers to claim here.
- AI operations depend on configured providers and can consume credits or paid model resources.
- A publication-ready, permission-safe product screenshot still needs to be captured.