

Blog ideation

Export: Blog stack ideation (Astro — Hyper vs Feed)

Summary decision

- Recommended base: **Engineering Blog with Series Support** (start from a starter and layer Tailwind).
- Two starter options explained: **Hyper** (code-heavy) and **Feed/Engineering-style** (series-centered).
- You can combine both: use Feed for series structure + import Hyper's code UX.

Key requirements (from you)

- Responsive (desktop & mobile)
- Markdown/MDX content
- Search (Meilisearch or Fuse.js fallback)
- SEO optimization + RSS
- Optional interactions (likes/comments)
- Unique look (use theme + Tailwind customization)

Starter theme options (recommended)

1. Hyper — excellent code blocks, dark/light toggle, copy buttons, MDX.
2. Feed / Engineering-style — series pages, next/prev, reading progress, series metadata.
3. Astro Blog starter — minimal MDX-ready base (fast to extend).

Chosen approach

- Start from Feed (series support) and import Hyper components for code UX.
- Add Tailwind for consistent design tokens and rapid customization.

Concrete tech stack

- SSG: Astro with MDX
- Styling: Tailwind CSS
- Search: Meilisearch (hosted) or Fuse.js (build-time JSON)
- Hosting: Cloudflare Pages
- Comments/Likes: Cloudflare Workers + KV/Durable Objects or Supabase (optional)
- RSS & SEO: Astro plugins/templates, sitemap.xml, schema.org

File/frontmatter schema (minimal)

- title, date, tags, category, series, part, difficulty, excerpt, featured_image, canonical

Immediate customizations to implement

- Series landing template with progress and next/prev
- Post template: large code blocks, copy button, reading time, difficulty badge
- Homepage: category lanes (Tech / Home Lab / Finance / Leadership)
- Search UI: category + difficulty filters
- Dark/light toggle and topic color tokens

10-day MVP roadmap (concise)

1. Init repo with Feed starter + MDX, add 3 sample posts
2. Add Hyper code components (copy button, tabs, syntax styles)
3. Install Tailwind; set tokens for topic colors & typography
4. Build series landing + post templates with frontmatter fields
5. Implement SEO/meta, sitemap, RSS
6. Implement search (Meilisearch or build JSON + Fuse.js)
7. Add optional interactions via Workers/Supabase (defer if MVP)
8. Optimize images, accessibility, responsiveness
9. CI/CD: connect to Cloudflare Pages, add build indexing step
10. Test & deploy; enable analytics

Meilisearch integration plan (brief)

- Index posts at build time via a script that reads MDX frontmatter and content, then pushes to Meilisearch (run during CI or separate indexing job).
- Expose search via client using a search-only API key or proxy through a Cloudflare Worker to hide keys.

Combination merge plan (Feed + Hyper) — 3 steps

1. Start Feed starter; keep series routing and templates.
2. Copy Hyper's code block components (syntax, copy button, theme toggle) into the Feed theme components.
3. Replace post code rendering to use the imported Hyper components and tweak Tailwind tokens.

Next actions I can do for you (choose one)

- Generate a starter repo scaffold (files + sample MDX posts + Tailwind config).
- Create the build-time Meilisearch indexing script and Cloudflare Worker proxy example.
- Produce exact package.json and Astro config to deploy on Cloudflare Pages.

Which export would you like me to produce as files/snippets?