

Raymond Camden Blog (raymondcamden2023) - Codebase Report

Overview / Purpose

This repository is the source for raymondcamden.com, Raymond Camden's long-running personal technical blog. The site publishes articles on web development, JavaScript, AI, ColdFusion, serverless, and related topics, along with static pages (About, Speaking, Now, Resume, etc.).

The project is a **static site** rebuilt on every deploy. It carries a large archive of historical content (posts dating back to 2003), legacy URL redirects from an older ColdFusion-era site, archived HTML comments from BlogCFC/Disqus, and several dynamic features layered on top via Netlify serverless and edge functions (search indexing, newsletter signup, recommendations, content negotiation).

Repository: github.com/cfjedimaster/raymondcamden2023

Primary Languages and Runtimes

Layer	Language / Runtime
Static site generator config & build logic	JavaScript (ESM) - Node.js >= 18 (required by Eleventy 3)
Templates	Liquid (primary), Nunjucks (.njk), EJS (.ejs)
Content	Markdown (blog posts), HTML
Netlify Functions	JavaScript / ESM (.js, .mjs)
Netlify Edge Functions	TypeScript (.ts, compiled for Deno edge runtime)
Client-side	Vanilla JavaScript (no React/Vue/Svelte app shell)

There is a `deno.lock` file used by Netlify's edge-function toolchain; it is listed in `.gitignore` and is not part of the committed build contract.

Frameworks and Major Libraries

Core static site stack

- [Eleventy \(11ty\) v3](#) - primary SSG; entry point is `eleventy.config.js`
- **Liquid** - default templating for layouts and most pages
- `@11ty/eleventy-plugin-ejs` - EJS support for a few listing pages
- `markdown-it + markdown-it-anchor` - Markdown rendering with auto-linked headings
- `@rknightuk/eleventy-plugin-post-graph` - visual post-frequency graph
- `eleventy-xml-plugin` - XML/RSS helpers
- `html-minifier` - HTML minification (CI-only transform)
- `clean-css` - CSS minification filter

Search & discovery

- **algoliasearch v5** - server-side indexing; client-side search on `/search/`

- Static JSON indexes generated at build time: `/algolia.json` (full) and `/algolia_new.json` (5 most recent posts)

Build-time data fetching

- `@11ty/eleventy-fetch` - cached remote fetches (e.g., Mastodon toots via `stoot` shortcode)
- `rss-parser` - RSS feeds (Letterboxd, Medium)
- Native `fetch` - Hardcover GraphQL, Untappd API

Serverless / edge (Netlify)

- `@netlify/blobs` - caches Algolia "related content" recommendations
- `@netlify/edge-functions` - content negotiation for Markdown/plain-text responses

Client-side (loaded from CDN or passthrough assets)

- `Alpine.js` - interactive search UI
- `Algolia Lite + Search Insights` - search page
- `Workbox 5` - service worker caching strategies
- `Giscus` - GitHub Discussions-based comments (2024+ posts)
- `GoatCounter` - privacy-oriented analytics
- `lite-youtube-embed`, `zero-md` - embed shortcodes

Authoring / maintenance scripts

- `@google/generative-ai` - Gemini-powered post summary/title suggestion scripts
- `dotenv`, `front-matter`, `axios`, `md5`

Architecture and Top-Level Structure

```

raymondcamden2023/
|-- eleventy.config.js      # Main Eleventy configuration (entry point)
|-- package.json           # npm scripts & dependencies
|-- netlify.toml           # Netlify headers & dev watch ignores
|-- guide.md               # Internal UI/shortcode reference for the author
|-- config/                # Extracted Eleventy modules
|   |-- collections/       # `posts`, `categories` collections
|   |-- filters/           # Liquid filters (escape, excerpt, cssmin, etc.)
|   |-- shortcodes/        # Shortcodes (excerpt, comments, YouTube, Mastodon, etc.)
|-- src/                   # Eleventy input directory
|   |-- _data/             # Global data (site.json + async API fetchers)
|   |-- _includes/         # Partial templates + 5,000+ legacy comment .inc files
|   |-- _layouts/          # default, page, post Liquid layouts
|   |-- assets/            # CSS, JS, images, PWA manifest, service worker
|   |-- misc/              # Algolia JSON, all-posts list, stats, racing data, etc.
|   |-- posts/             # Blog archive (YYYY/MM/DD/*.md) - ~6,796 files total
|   |-- *.liquid/html/md   # Top-level pages, feeds, sitemap, search, etc.
|-- netlify/
|   |-- functions/         # Serverless API endpoints
|   |-- edge-functions/    # Content negotiation edge function
|-- scripts/               # Maintenance: genpost, Algolia rebuild, AI helpers
|-- .agents/skills/        # Cursor agent skills (precheck, prepublish)

```

```
`-- _site/                                # Build output (gitignored)
```

High-level request flow

```
flowchart TD
  A[Visitor request] --> B{Netlify Edge}
  B -->|Accept prefers text/markdown| C[Fetch /path/index.md mirror]
  B -->|Normal browser| D[Static HTML from _site]
  D --> E[Client JS: search, SW, analytics, Giscus]
  A --> F[/api/* Netlify Functions]
  F --> G[Buttondown / Algolia / Blobs / GoatCounter]
```

Content model

Blog posts live under `src/posts/YYYY/MM/DD/YYYY-MM-DD-slug.md` with front matter like:

```
layout: post
title: "... "
date: "2026-08-11T18:00:00"
categories: ["development"]
tags: ["web components", "javascript"]
banner_image: /images/banners/files.jpg
permalink: /2026/08/11/file-preview-a-simple-web-component-for-file-previews
description: ...
```

The `getPosts` collection mutates each post at build time to append `.html` to permalinks and normalize category casing.

Build-time vs runtime split

Concern	Where it runs
Page HTML, feeds, sitemap, Algolia JSON	Eleventy build
Hardcover books, Letterboxd films, Untappd beers, Medium RSS	Eleventy <code>_data</code> fetchers (skipped locally via <code>SKIP_REMOTE_DATA=1</code>)
Mastodon embeds (<code>{% stoot %}</code>)	Async shortcode with 2-week Eleventy Fetch cache
Newsletter signup	Netlify Function -> Buttondown API
Post-deploy Algolia partial reindex	<code>deploy-succeeded</code> function
Related posts	<code>get-recommendations</code> function + Netlify Blobs cache
Markdown alternate representation	Edge function + <code>eleventy.after</code> hook copies posts to <code>output/YYYY/MM/DD/slug/index.md</code>
Search UI	Client-side Algolia against public search-only key

Local vs production build scope

- **Production (Netlify):** builds all ~**6,796** posts (no `.eleventyignore` in git - that file is gitignored and exists only locally).
- **Local dev:** `.eleventyignore` excludes posts before 2024, yielding ~**357** posts for faster iteration. Combined with `SKIP_REMOTE_DATA=1`, local builds avoid live API calls.

Entry Points and How to Run, Build, and Test

Install

```
npm install
```

Development

Command	What it does
npm run dev / npm start / npm run serve	SKIP_REMOTE_DATA=1 netlify dev -c "eleventy --serve --quiet" - Netlify CLI proxy + Eleventy dev server
npm run dev2	Eleventy only (no Netlify function/edge emulation): SKIP_REMOTE_DATA=1 eleventy --serve --quiet

Netlify dev exposes serverless functions under /api/* and edge functions locally.

Production build

There is **no build script** in package.json. Production builds are expected to run via Netlify (typically npx eleventy), with build settings configured in the Netlify site dashboard rather than in netlify.toml.

Manual build:

```
npx eleventy
```

For a local smoke test matching agent-skill conventions:

```
SKIP_REMOTE_DATA=1 npx eleventy
```

Output goes to _site/ (Eleventy default).

Post authoring

```
node scripts/genpost.js "My Post Title" '["development"]' '["javascript"]'
```

Creates a dated Markdown file under src/posts/YYYY/MM/DD/.

Maintenance scripts (manual, not npm scripts)

Script	Purpose
scripts/rebuilddalgolia.js	Full Algolia reindex from /algolia.json (CommonJS + dotenv)
scripts/summarysuggestion.js	Gemini summary suggestion for a draft post
scripts/titlesuggestions.js	Gemini title suggestions
scripts/list-orphan-comments.mjs	Finds archived .inc comment files with no matching post

Testing

No automated test suite is present - no Jest/Vitest/Mocha config, no test scripts, no GitHub Actions test workflow. Validation is manual plus two Cursor agent skills:

- **precheck** - reviews engine/infrastructure changes; runs `SKIP_REMOTE_DATA=1 npx eleventy` when config/templates change
 - **prepublish** - spell-check and content review for new blog posts
-

Notable Tooling

Package manager

- **npm** (`package-lock.json`)

Hosting / deploy

- **Netlify** - static hosting, functions, edge functions, redirects, headers
- **deploy-succeeded** webhook function - partial Algolia reindex (3 newest posts) + Pushbullet deploy notification
- **deploy-status** - queries Netlify deploy API for last deploy state

CI / linting

- **No GitHub Actions workflows** - only `.github/FUNDING.yml`
- **No ESLint, Prettier, or Stylelint** configuration in the repo
- **HTML minification** runs only when `process.env.CI` is set (Netlify build environment)

Analytics & monetization

- **GoatCounter** (script in default layout)
- Content Security Policy allows Carbon Ads and Infolinks
- Buy Me a Coffee / Patreon CTAs in post footer

PWA

- `src/assets/manifest.json` - web app manifest
- `src/assets/service-worker.js` - Workbox NetworkFirst caching for `/`, JS, CSS, images, and post URL patterns

Redirects & SEO

- `src/_redirects` - hundreds of legacy URL 301 redirects from old ColdFusion paths
 - `src/sitemap.njk` - XML sitemap from `collections.all`
 - `src/llms.txt` - LLM-oriented site summary (recent posts + nav links)
 - `src/webfinger.liquid` - Mastodon WebFinger at `/.well-known/webfinger`
 - Open Graph images via Cloudinary transforms and Eleventy Screenshot service for homepage
-

How the App Works - Notable Details

Page types

Route / feature	Source
/	src/index.liquid - hero + 6 latest posts
/all/	src/misc/all.liquid - collapsible year archive
/search/	src/search.liquid - Algolia + Alpine.js
/now/	src/now.liquid - live reading/watching/listening/beer data
/feed.xml, /feed_slim.xml	Atom feeds (full vs mailing-list-trimmed)
/categories/*, /tags/*	Paginated Eleventy collections
/postgraph/	Post frequency visualization
/llms.txt	AI crawler summary
Blog posts	src/posts/** -> post.liquid layout

Comments (three systems)

1. **Legacy archived HTML** - 5,110 .inc files under src/_includes/comments/, loaded by hasAnyComments / commentInclude shortcodes with flexible date-path matching
2. **Giscus** - GitHub Discussions embed for posts from 2024 onward
3. **Disqus include** - mydisqus.liquid still referenced but effectively superseded for new content

Search pipeline

1. Build generates /algolia.json (all posts) and /algolia_new.json (5 newest)
2. On deploy success, deploy-succeeded pushes the 3 newest entries to Algolia
3. Full reindex available via scripts/rebuildalgolia.js
4. /search/ queries Algolia client-side with a public search-only API key

Content negotiation (Markdown mirror)

After each build, an eleventy.after hook copies built posts to /_site/YYYY/MM/DD/slug/index.md. The edge function content-negotiation.ts serves these when Accept prefers text/plain OR text/markdown over HTML.

Remote data on the Now page

Data	Source	Env var
Books	Hardcover GraphQL	HARDCOVER_BOOKS
Films	Letterboxd RSS	(none)
Beers	Untappd API	UNTAPPD_CLIENT_ID, UNTAPPD_CLIENT_SECRET
Music	Pipedream endpoint (client-side fetch)	hardcoded URL in now.liquid

All _data fetchers respect SKIP_REMOTE_DATA=1 for local builds.

Netlify API endpoints

Path	Function	Purpose
/api/newsletter-signup	newsletter-signup.js	Subscribe via Buttondown
/api/get-recommendations	get-recommendations.mjs	Algolia related posts (12h Blob cache)

/api/get-stats	get-stats.mjs	Buttondown subscriber count + GoatCounter top pages
/api/get-cats	get-cats.js	Joke cat JSON API (demo content)
deploy-succeeded	webhook	Algolia partial update + Pushbullet notification
deploy-status	polling	Last Netlify deploy info

Shortcodes & author tooling

Documented in `guide.md` and implemented in `config/shortcodes/`:

- `{% callout %}`, `{% mdwrap %}`, `{% liteyoutube %}`, `{% darkgist %}`
- `{% stoot instance id %}` - embedded Mastodon posts
- `{% excerpt post %}`, comment includes

Global data

`src/_data/site.json` defines site title, description, navigation, and social profile links used across layouts and `llms.txt`.

Material Observations and Trade-offs

Scale: ~6,800 posts plus 5,100 comment include files make full local builds slow; `.eleventyignore` is a local-only workaround and is not committed, so production always pays the full build cost.

No formal CI/tests: Engine changes rely on manual Eleventy smoke builds and agent skills rather than automated regression coverage - reasonable for a personal blog, but risky for search/indexing or redirect logic changes.

Mixed module systems: Most of the repo is ESM (`"type": "module"`), but `scripts/rebuildalgolia.js` and `scripts/summarysuggestion.js` use CommonJS - a maintenance footgun called out in the precheck skill.

Secrets split: Build-time data uses env vars; the search page embeds a public Algolia search key in client HTML (expected for search-only keys, but worth auditing periodically).

Duplicate dependencies in `package.json` (`markdown-it-anchor`, `@google/generative-ai`, `@rknightuk/eleventy-plugin-post-graph`, `front-matter` listed twice) - harmless but sloppy housekeeping.

Third-party coupling: Newsletter (Buttondown), comments (Giscus/GitHub), search (Algolia), analytics (GoatCounter), images (Cloudinary/static.raymondcamden.com), and several external APIs on the Now page - the site is static at its core but operationally dependent on multiple SaaS integrations.