WEEK 6 JOB LIFE

Add a Blog System

Learning Objectives

By the end of this session, students should be able to:

- **Graft a real blog onto an existing Next.js app** by describing the feature to Cursor and pointing it at a reference template — without ever hand-editing route files.
- **Write and publish your first MDX post** so that it renders on your live site with proper typography, a cover image, and a working permalink.
- **Ship a valid RSS feed** that a feed reader can subscribe to, because shipping a blog without an RSS feed is shipping half a blog.

Core Topics

- MDX = Markdown + React components. Why this is the right content format for a developer blog.
- Routing: how `app/blog/[slug]/page.tsx` maps to a URL.
- Metadata patterns: frontmatter, OG images, canonical URLs, reading time.
- RSS in 2026: still the cleanest way to be findable and syndicated.
- The "reference-template" prompt pattern: teaching Cursor from an existing good example instead of re-inventing from scratch.

Tools / Stack

Tool	Role this week
Vercel Blog Starter	The reference template we point Cursor at.
MDX	Your post files — Markdown + React components.
`remark-gfm` + `rehype-pretty-code`	Syntax-highlighted code blocks in posts.
`reading-time` + `gray-matter`	Frontmatter + reading-time utilities.
`feed` **(npm)**	Generates valid RSS 2.0 / Atom feeds at build time.

Your personal site — cumulative features through this week:

Wk 1 · Dev toolkit installed — Cursor, Claude Code, Gemini CLI, MCPs, Typst skill

Wk 2 · Personal website deployed to Vercel

Wk 3 · AI chat widget (Gemini 2.5 Flash) emulating you

Wk 4 ·

Neon Postgres + Neon Auth + Drizzle guestbook

Wk 5 · Vercel Blob image uploads in guestbook

Wk 6 · **MDX blog system with RSS** ← NEW

Session Plan

Time	Activity
0 – 15 min	**Recap & Check-in.** Any memes from the guestbook go viral this week? 30 seconds of show-and-tell.
15 – 40 min	**Concept Teaching.** MDX vs plain Markdown. Why route groups help keep `/blog` separate from the rest of the site. Static vs dynamic rendering — for a blog, static wins.
40 – 75 min	**Live Demo.** Instructor adds `/blog` to her site by referencing the Vercel blog starter, then writes a post **"What I learned in weeks 1–5"** in real time. Class watches typography and RSS both land in 15 minutes.
75 – 105 min	**Hands-On Lab.** Students graft `/blog` onto their sites and publish their first real post — recap of the course so far, written in their voice.
105 – 120 min	**Q&A + Wrap.** Three students read their post titles aloud; class picks the most intriguing.

Hands-On Lab

Task. By the end of class your site has `/blog` with at least one published post — your own first-person recap of weeks 1–5 — plus an RSS feed at `/feed.xml` that validates.

PROMPT **Step 1 · Say to Cursor:**

> *Please add a blog system to my existing `my-portfolio` project. Use the Vercel Next.js Blog Starter template as a reference for structure — specifically the way it organises `app/blog`, MDX components, and post loading. Don't overwrite my existing site; **graft** the blog in.*
>
> *Specifically:*
>
> 1. *Create `app/blog/page.tsx` — the index listing all posts, newest first.*
> 2. *Create `app/blog/[slug]/page.tsx` — individual post pages.*
> 3. *Create `content/posts/` as the directory where MDX files live.*
> 4. *Install and wire MDX support: `@next/mdx`, `remark-gfm`, `rehype-pretty-code`, `gray-matter`, `reading-time`.*
> 5. *Add a **Blog** link to the site header next to my existing nav items, matching the current design language.*
>
> *Don't write any posts yet. Just give me the plumbing + an empty listing page that says "No posts yet" when there are zero MDX files.*

VERIFY **Step 2 · Verify:**

- Navigating to `/blog` on localhost shows the empty state.
- The header link **Blog** is visible and matches the site's other links.
- `content/posts/` exists and is empty.
- `package.json` has the new MDX dependencies.

PROMPT **Step 3 · Say to Cursor:**

Now polish the post page template before I write real content. Requirements:

- *Post layout uses a readable measure (~68 ch). Serif body type for reading.*
- *Support frontmatter:* `title`, `date` *(ISO),* `summary`, `cover` *(optional image path in* `public/`*),* `tags` *(optional array).*
- *Header renders: cover image (if present), title (H1 large), date + reading-time*
  1. *tags on one muted line.*
- *Code blocks get one-dark theme via* `rehype-pretty-code`*.*
- *Images inside posts use Next.js Image with lazy-load + blur placeholder.*
- *At the bottom of every post: a small* ***Written by [my name] — part of the TECHNEST AI Programming track*** *byline linking back to* `/`*.*
- *Post pages export proper metadata for Open Graph so shares on Twitter/WeChat look good.*

VERIFY **Step 4 · Verify:**

- Create a dummy MDX file in `content/posts/` with all the frontmatter fields populated; view it at `/blog/\[slug\]`.
- Typography is pleasant; code blocks are highlighted; cover image shows.
- View source: `<meta property="og:title">` matches the post title.

PROMPT **Step 5 · Say to Cursor:**

Let's write my first real post. Draft `content/posts/weeks-1-to-5-recap.mdx` *as an honest first-person recap of what I built in the last five weeks. Here's my raw input — please polish it into ~500–700 words in my voice (match the tone of my About page and persona.md).*

• Week 1: first AI-driven Vercel deploy. What surprised me was [ADD YOUR OWN TAKE HERE BEFORE ASKING CURSOR]. • Week 2: Magic Portfolio template. Unexpectedly satisfying detail: [YOURS]. • Week 3: AI-clone chat. Funniest thing it said about me: [YOURS]. • Week 4: Neon + auth. Biggest moment of confusion: [YOURS]. • Week 5: meme uploads. Best meme so far: [YOURS].

Pick a good, specific title — not "My Journey". Tag the post with ***technest*** *and* ***learning****. Set a cover image (reuse the hero photo from my About page). Once written, paste the full Markdown back to me so I can read and tweak before committing.*

VERIFY **Step 6 · Verify:**

- Read the draft. Tweak any sentence that doesn't sound like you.
- Ask Cursor to save the final version and commit it.
- Refresh `/blog` — your post appears in the index.
- Click through — it reads well, cover image loads, byline link works.

PROMPT **Step 7 · Say to Cursor:**

> *Add an RSS feed at `/feed.xml` that includes all posts (full content, not just excerpts — RSS readers prefer full content). Use the `feed` npm package. It should:*
>
> - *Auto-generate at build time from the MDX files.*
> - *Include: title, author, canonical link, pubDate, summary, full HTML content, categories from the tags array.*
> - *Validate as RSS 2.0 (test against `https://validator.w3.org/feed/`).*
> - *Add a `<link rel="alternate" type="application/rss+xml" ...>` tag to the site's `<head>` so browsers and RSS readers can discover it automatically.*

VERIFY **Step 8 · Verify:**

- `curl https://localhost:3000/feed.xml` returns well-formed XML (Cursor will run this).
- Open the URL in your browser — you see the feed content.
- View the page source of your homepage — the `<link rel="alternate">` is there.

PROMPT **Step 9 · Say to Cursor:**

> *Ship it: commit all the new files in logical chunks, push to GitHub, wait for the Vercel deploy, then verify on the live URL. Also: submit the live `/feed.xml` URL to the W3C feed validator and report back.*

VERIFY **Step 10 · Verify:**

- Production site has `/blog` with your post.
- `/feed.xml` on the live site validates clean.
- Paste your live post URL into WeChat / Slack / Twitter — the preview card shows title + cover + summary correctly.

RECOVER **Step 11 · If stuck, say to AI:**

> *The RSS feed shows the post summary but the `<content:encoded>` body is empty. Can you figure out where MDX-to-HTML conversion is missing in the feed generator and fix it so the full post body appears in the feed?*

TIP · Write the post before polishing the template

A common mistake: spending 90 minutes on typography and 5 minutes on content. **Content first.** Your classmates will remember what you wrote, not your letter-spacing.

CAREER · Blogging compounds quietly

Four to five posts over this term is already more public writing than 90% of engineers ever publish. One year later, a recruiter searching **"agentic systems learning in public"** might find your post and DM you. Compound effort.

Weekly Assignment

Build / Implement.

- `/blog` on your live site with at least **one real post** (~500+ words, your own voice, cover image).
- Valid `/feed.xml` RSS feed.
- Blog link in the site header.

Requirements.

- MDX support (not plain Markdown — must import a React component into at least one post to prove the pipeline works).
- W3C RSS validator passes clean on the production feed URL.
- OG image renders correctly when you paste the URL into a chat app.

Submission. Live post URL + RSS validator pass screenshot, in Slack before Week 7.

Resources

Docs

- Vercel Blog Starter — reference template
- `@next/mdx` — App Router integration
- `rehype-pretty-code` — syntax highlighting
- W3C Feed Validation Service

Videos

- Instructor demo: "Blog + RSS in 15 minutes"

Repos

- `vercel/examples/solutions/blog`
- `shadcn-ui/taxonomy` — a more opinionated blog example

Real-World Application

Every engineer you respect has a blog. Not a "hire me" blog — a **thinking-out-loud** blog. This week you built the infrastructure. The habit of writing one post per week, even for just two months, will change your career more than any certification. Your future self will send DMs of thanks to your present self for starting today.

> CAREER
>
> Cross-post your first piece to LinkedIn and dev.to manually. After that, you can automate (we'll do that in a bonus week if time allows). The first manual cross-post teaches you what parts of the post make people click.

Challenges & Tips

- **"MDX compile error: can't parse frontmatter."** Frontmatter must be the very first thing in the file, no blank line before. Ask Cursor **"Fix my MDX frontmatter so it parses cleanly."**
- **"My cover image is huge and slow to load."** You're using the original 4 MB photo. Ask Cursor **"Pre-optimise blog cover images at build time to 1200 × 630 for OG, and 800 × 450 for the post page."**
- **"The RSS feed validates but Reeder shows nothing."** Your `<link>` tag in the head is probably self-closing incorrectly. Ask **"Check the RSS autodiscovery link tag in my `<head>`."**

- **“Post page shows `next/headers` error.”** You used a Server-only function in a Client component. Ask Cursor **“Separate the Client parts of the post page into a `ClientPost` component.”**
- **“The Tags link in the index goes to /blog/tag/foo but renders 404.”** You promised dynamic tag pages but haven’t implemented them. Either ask Cursor to add `/blog/tag/[tag]/page.tsx`, or remove the tag-link behaviour for now.

> **STUCK? RESCUE**
>
> If a post renders locally but not in production, check for case-sensitive filenames. macOS is case-insensitive, Vercel’s Linux is not. Ask Cursor to **“normalise all MDX filenames to lowercase and update any imports accordingly.”**

Instructor Notes (Internal)

- **The 500-word first post is surprisingly hard for most students.** They freeze. Walk the room during step 5 and unstick anyone with a direct question: **“What’s one specific thing that confused you in week 4 and got clear in week 5?”** — then tell them **“that paragraph is your post.”**
- **Don’t over-spec the typography.** Cursor will default to tailwind-typography (`prose`) which looks fine. Students can iterate later.
- **RSS discovery matters more than RSS itself.** Make sure the `<link rel="alternate">` lesson lands — most students don’t know this is how feed readers autodiscover feeds.
- **Pre-populate the class reading list.** Paste the links to the best blogs by engineers the students should read (Alice Maz, Dan Abramov, Julia Evans, Stripe Press blog). Gives concrete taste to imitate.
- **Quiet moment:** at the end of class, have everyone subscribe to three classmates’ RSS feeds. Makes the effort feel shared.

Student Feedback

Challenges Observed

Extra Information

MEET YOUR INSTRUCTOR

Chan Meng

Full-Stack Engineer · AI Agent Builder · Minimalist

> *"Subtraction for life, addition for thought."*

AI AGENT BUILDER

AGENTIC SYSTEMS · LLM INTEGRATION

FEMTECH INNOVATOR

MINIMALIST CODE · UN CSW 69 SPEAKER

Chan builds **production-grade AI agents** at the intersection of **agentic systems, LLM integration, and women's health**. Between China and New Zealand, she ships systems that go to work for real users at real companies — and she believes code, like life, becomes most powerful when you subtract what isn't essential. A **Master of Applied Computing with Distinction** from Lincoln University, NZ (Dean's List), she was featured at **UN Women CSW69** for her work on AI and gender equality.

Currently · Founding Engineer at Gavigo · Senior Full-Stack Engineer at She Sharp (NZ's leading women-in-STEM nonprofit, 2,200+ members) · CTO of FemTech Weekend · Open Source Contributor to CopilotKit (24.6k★). For the full project portfolio and code, visit github.com/ChanMeng666.

Stay in the loop

NEWSLETTER · RECOMMENDED

"What's Shipping in AI" — a weekly curated digest, every Monday. chanmeng.org/#newsletter

LINKEDIN
chanmeng666

GITHUB
ChanMeng666

PORTFOLIO
chanmeng.org

EMAIL
chanmeng.dev@gmail.com

Building tools that matter. Let's connect.