Section renderer checklist

For any SvelteKit page built from CMS sections. Each check comes from a measurement in the free demo repo (link at measuredfrontend.com/free).

Is the content in the HTML?

- [ ] **Lazy sections are resolved before the server renders.** Components come from the universal load (`+page.ts`), not from `{#await import()}` in the template and not from `onMount`. *In the demo those two left 3 of 5 sections out of the server HTML.*
- [ ] **You checked with curl, scripts stripped.** `curl -s URL | perl -0pe 's/<script.*?<\/script>//gs' | grep -c "text from a lazy section"` returns more than 0. *Without stripping scripts, the serialized page data gives a false match.*

Does each page download only what it shows?

- [ ] **Small core map, everything else lazy.** Sections most pages start with are imported directly; the rest go through `import.meta.glob`. *Static imports made a 2-section page download code for 3 unused sections.*
- [ ] **Unknown types render nothing.** The type is looked up in a fixed map; no import path is built from CMS data.

Does one bad entry stay one bad section?

- [ ] **CMS data is validated in the server load.** One rule per section type that can crash on bad props; invalid sections are dropped and logged. *One Gallery with no images returned a 500 for the whole page without it.*
- [ ] **You don't rely on `<svelte:boundary>` during SSR.** *A section that threw during server rendering returned a 500, boundary or not (Svelte 5.57).*
- [ ] **You know what happens when a lazy chunk fails.** *With the chunk blocked, the server-rendered section disappeared at hydration.* Sections that must never disappear belong in the core map.

Bonus

- [ ] Heavy sections are a hydration cost for the **whole page**: nothing hydrated until the slowest server-rendered section's code arrived.

Want the rules in your AI assistant? Measured Rules turns these checks into rules for Claude Code, Cursor, Copilot and AGENTS.md, each backed by a passing test. measuredfrontend.com/rules/lazy-cms-sections