1. git init
2. git clone https://github.com/tryretool/custom-component-collection-template new-custom-component
3. cd new-custom-component
4. Replace existing index.tsx with the given index.tsx file.
5. npm install
6. npx retool-ccl login
7. It will be required to provide subdomain name and api key. You can create that from Retool app in settings.
8. npx retool-ccl init
9. npx retool-ccl deploy
10. Go to the Retool and you will be able to find that newly created custom block, drag it to the canvas and after selecting that click on the inspector, there you will be able to see the input fields( API key, prompt, options equivalent as compliance frameworks + action)
11. The outputs (redacted\_prompt, risk\_score, violations) will be shown inside the block, and to access these in retool app it can be used as {{ myBlockId.redacted\_prompt }}, {{ myBlockId.risk\_score }}, and {{ myBlockId.violations }}.

Quick Start Guide

1. After performing above steps.

3. In the Inspector pane of the block, you’ll see four input fields:

• apiKey → Provide your PromptLock key

• text → The prompt you want analyzed

• compliance\_frameworks → JSON array or comma-separated list of frameworks ( ["gdpr","pci"])

• action\_on\_high\_risk → e.g. "redact"

4. The block’s outputs (redacted\_prompt, risk\_score, and violations) are automatically exposed.

• Access them in Retool as {{ yourBlockName.redacted\_prompt }}, etc.

5. Try a test prompt (e.g., “John Smith’s SSN is 123-45-6789.”).

6. The block calls https://promptshield-g1cz.onrender.com/v1/analyze. If successful, you will see the sanitized “redacted\_prompt,” numeric “risk\_score,” and an array of “violations” in JSON.

7. Optional: Map the redacted\_prompt to an OpenAI resource if you want to pass sanitized text to GPT.