

Level 2: Applied

All Librarians

Module 10

Prompt library for library work

The difference between occasional AI users and power users is a prompt library. This module teaches you to build one, systematically capturing what works so you don't start from scratch every time.

By Yulia Brusova

20 min read · Reviewed October 2026

AI for Academic Libraries · ai-for-academic-libraries.vercel.app

© 2026 · Licensed under CC BY-NC-SA 4.0

WHAT YOU'LL BE ABLE TO DO

- 1 Build a personal prompt library with at least ten reusable prompts for your library work
- 2 Set up a Claude Project (or equivalent) with persistent context for a recurring workflow
- 3 Share prompts effectively with colleagues so they can adapt them for their own use
- 4 Explain why a prompt library reduces variability in AI output
- 5 Develop a lightweight process for adding to and improving your prompt library

The most consistent pattern among librarians who use AI effectively is not better tools or more time to experiment. It is that they have stopped starting from scratch. The first time a prompt produces genuinely useful output, most people note that it worked and move on. The next time they need something similar, they reconstruct the prompt from memory, get a different result, and spend time adjusting. A prompt library is the professional practice of not letting that happen. It means systematically capturing what works, organizing it for retrieval, and building on it over time instead of rediscovering the same effective approaches. The investment required is modest. The return, compounded across every task where you would otherwise start from scratch, is substantial.

What a prompt library is and why variability is the problem it solves

A prompt library is a curated collection of prompts, with the contextual notes that make them reusable, organized for recurring professional tasks. It is not a log of every conversation you have had with an AI tool. Nor is it a repository of prompts you have tried and might try again. It is a set of prompts that have been tested, have produced consistently useful output, and are documented well enough to reuse without reconstructing them from memory.

The problem a prompt library solves is variability. AI output quality is not fixed. It varies significantly with how the prompt is constructed, what context the model has, and what constraints the prompt imposes. A librarian who drafts reference email responses by typing a quick request each time will get different quality on different days. The result depends on how much context they happen to include, how clearly they specify tone and length, and how

well they articulate the task in the moment. A librarian who runs the same inquiry through a tested, context-rich prompt template gets consistent output that they edit rather than rewrite. That difference, between inconsistent outputs that need substantial revision and consistent outputs that need light editing, is the practical consequence of working with or without a prompt library.

The analogy that resonates most with librarians is the reference interview. A reference librarian with a reliable set of clarifying questions for common inquiry types does not reconstruct them for each patron. They have internalized an approach that reliably produces the information they need. A prompt library externalizes the same logic: the "clarifying questions" are built into the prompt structure, and the librarian applies a tested approach rather than improvising each time. A reference interview framework also improves with use, keeping questions that consistently produce useful responses and revising those that cause confusion. In the same way, a prompt library improves as the librarian learns which prompt structures work reliably in their professional context.

What makes any prompt work

The prompts in this module are starting points, not formulas; adapt them to the specific task and context each time. What makes any prompt work is what makes any professional communication work: adequate context, clear expectations, and an honest account of what is actually needed. Dakan and Feller identify a consistent pattern across effective AI prompts. They specify not just what to produce, but who the output is for, what constraints apply, and how the AI should approach the task (Dakan & Feller, *AI Fluency: Framework & Foundations*, Anthropic Academy, 2025). The most common reason a prompt underperforms is that it named the desired output but not the purpose or audience. For example, the difference between asking for "a bibliography on food insecurity" and asking for "five annotated sources on food insecurity appropriate for a first-generation college student writing a persuasive essay" is the difference between a usable result and an editing project.

When a prompt is not producing what is needed, the most efficient move is often to ask the AI directly to help improve it. Describe the actual goal and ask for a better way to phrase the request. In practice this sounds like: *"I'm trying to create a research guide for students who have never used a database, but I'm not getting what I need. Can you help me ask this better?"* This treats the AI as a thinking partner in the communication task itself, which is often faster than iterating on the prompt alone.

What belongs in a library prompt library: categories and examples

Do not start by creating prompts for every possible task. Identify the five or six tasks you perform most often where AI assistance has already proven useful, and build tested prompts for those first. A focused beginning produces a small library that is actually used, rather than a comprehensive one consulted occasionally. The library then grows naturally as you meet tasks where a saved prompt would have saved time.

For most academic librarians, the highest-value categories fall into four areas. The first is patron communication: reference email responses by inquiry type, FAQ content for specific services, and subject guide descriptions for specific disciplines and patron populations. A tested reference email prompt specifies institution type, patron population, preferred tone, a length constraint, and what to include or avoid. With only the patron's question swapped in, it will produce a useful draft for the vast majority of inquiries. For example: "You are a library associate at a community college serving a high proportion of first-generation students. A student has sent the following reference inquiry: [paste question]. Write a 150-word email response in a warm, encouraging tone that directly addresses the question, points the student to the most appropriate starting resource, and offers to follow up. Do not use library jargon." Once such a prompt is tested and confirmed to need only light editing, save it with that exact phrasing.

The second category is instruction: lesson plan drafts, learning objective sets, active learning activity ideas, and assessment questions. A tested lesson plan prompt specifies session length, patron population, course context, the assignment being supported, and the learning outcome to prioritize. It will produce a starting structure for almost any single-shot instruction session. The third category is administrative writing: annual report sections, grant narrative segments, policy draft frameworks, and meeting agendas. These tasks are all time-consuming, structurally familiar, and amenable to AI drafting with light human review. The fourth category is research support: search strategy development for specific disciplines, document summarization in a specified output format, and synthesis across multiple sources for a stated purpose.

A practical rule for deciding what to add: if you used a prompt and thought "I should save that," save it within the same session. If you wait, the exact wording is usually gone, and you have only a general memory of what worked. To make saving effortless, keep the library somewhere you can reach immediately (a pinned document, a saved note, a dedicated tab), not in a system that takes several navigation steps before you can add an entry.

Anatomy of a reusable prompt: variables, context, and constraints

The structural difference between a single-use prompt and a reusable one is the presence of variables. A reusable prompt has clearly marked placeholders for the elements that change from task to task. It also states explicitly the context and constraints that stay constant across uses. A single-use prompt is optimized for one task in one moment; a reusable prompt is optimized for a class of tasks and documents the parameters that make it work.

A reusable library prompt has four standard components. First, a role or context specification that tells the model who it is and what professional setting applies: "You are a reference librarian at a midsize public university library serving undergraduate and graduate students." This does not need to change between uses; it is the constant context that orients every output. Second, a task description with bracketed variables for the elements that change: "A [patron type] has sent the following inquiry: [paste inquiry]. Draft a [length]-word response in a [tone] tone." Third, a constraint set specifying what the output should include or avoid: "Include a recommended starting database, a brief explanation of why it is appropriate for this inquiry, and an offer to follow up. Do not use library jargon or assume prior database experience." Fourth, an output format specification when the format matters: "Format the response as a complete email, beginning with a greeting and ending with your name and title."

The bracketed variables ([patron type], [inquiry], [length], [tone]) are where the prompt is adapted for each use. Everything else stays constant. For example, the reference email prompt above might be used for an undergraduate research inquiry, a graduate thesis consultation, and a faculty course reserve request in the same week. Only the patron type and the pasted inquiry change between uses. Keeping the non-variable components constant is what produces consistent output quality. The model receives the same professional context, constraint set, and format specification each time. Variation in the output then reflects genuine variation in the task, not in how well the prompt happened to be written that day.

A practical note on prompt length: more specific prompts produce more consistent output, but beyond a certain point, adding constraints yields diminishing returns. Prompts in a well-functioning library are typically longer than what a librarian would write on the spot, with more context, more explicit constraints, and more format guidance. They incorporate everything learned from testing about what produces reliable output. To find the right level of specification for a task, start with a reasonable prompt and test it on three or four

representative examples. Note where the output needed the most editing, and revise the prompt to address those failure modes. This iterative refinement (test, identify failures, revise) is what distinguishes a tested prompt from a drafted one.

Claude Projects and persistent context: setting up your professional workspace

A prompt library stores what to ask; persistent context stores who is asking and why. The two work together. A librarian with a library of reference email prompts still benefits from a persistent context setup. With it, they never have to restate their institution type, patron population, and professional role in every prompt, because the model already has that from the project context. This combination, a library of tested task-specific prompts used within a persistent professional context, produces the highest-quality outputs with the least per-session setup.

Claude Projects is the current implementation of persistent context in Claude. A project is a defined workspace with its own conversation history, custom instructions that apply to every conversation within it, and uploaded documents the model can reference throughout. For example, a reference and instruction librarian might set up a project with custom instructions that include: "I am a reference and instruction librarian at a community college with a large first-generation and returning adult student population. My institution has limited database subscriptions: primarily JSTOR, Academic Search Complete, and ProQuest. My patron-facing communication should always use plain language, avoid library jargon, and assume students may be anxious about library research." Set once and applied to every conversation in the project, these instructions remove the need to re-establish professional context at the start of every session.

Uploaded documents within a project extend the persistent context to include institution-specific reference materials. Suppose a librarian uploads the library's current database list, its hours and services FAQ, its information literacy outcomes document, and two or three of their strongest lesson plans. That creates a knowledge base the model draws on in every conversation, without the librarian pasting that content into each prompt. For example, when drafting a subject guide for a nursing course, the model can check the uploaded database list for what the institution actually subscribes to, instead of suggesting databases the library lacks. Grounding outputs in institutional reality this way is a significant improvement over generic outputs based on training data alone.

ChatGPT's Custom Instructions and Custom GPTs serve a similar function, with different architectures. Custom Instructions apply persistent context across all conversations, which suits general professional context. Custom GPTs create a distinct configured assistant with specific instructions, uploaded knowledge, and defined capabilities, which suits a specific recurring workflow. For team use, Custom GPTs can be shared within an organization's ChatGPT Business or Enterprise plan. That matters for libraries that prioritize prompt sharing and collaborative workflow development. The practical guidance for most library professionals is to use whichever persistent context mechanism exists in the tool they use most. The architecture differs, but the value of not restating your professional context in every session is the same on every platform.

Sharing prompts across your library: multiplying individual effort

A prompt that works for one librarian will work for colleagues, with adjustments for their specific context. The institutional value of a prompt library is not fully realized until the prompts are shared. An individually maintained library represents one person's accumulated knowledge. A shared library represents the team's, and it is substantially more valuable than any individual contribution to it.

The simplest sharing mechanism is a shared document, such as a Google Doc or Notion page, organized by task category. Each prompt should appear in a standard format: the prompt text, a single-sentence description of the output it produces, the task it is designed for, notes on adjusting it for different contexts, and the AI tool it was developed and tested on. The format matters because prompts shared without context are rarely adopted. A colleague who sees a block of prompt text with no explanation of when to use it or what it produces has no basis for judging its relevance. Now suppose the same prompt carries a note: "Use this for drafting reference email responses to undergraduate research inquiries; it produces a 150-word draft in an accessible tone that typically needs only minor edits." The colleague can immediately judge whether it fits their workflow and how to adapt it for their institution.

The LibGuide format is particularly well suited to institutional prompt sharing. It is familiar infrastructure that library staff already know how to navigate, and its tabs and boxes let prompts be organized by task type for browsing rather than keyword search. A staff-only LibGuide page for library AI prompts (visible to colleagues but not to patrons) is a natural home for a shared prompt library that needs no new technology. For example, a LibGuide with tabs for Patron Communication, Instruction, Administrative, and Research Support

prompts, each prompt in the standard format described above, is immediately usable by colleagues who never contributed to it.

The social dimension of prompt sharing deserves direct attention. Prompts are professional knowledge. A well-crafted reference email prompt represents accumulated expertise about what makes patron communication effective in a specific institutional context. To make sharing feel like contribution rather than extraction, acknowledgment matters: note who developed a prompt, invite colleagues to submit prompts they have developed, and treat the shared library as a team resource rather than an anonymous collection. Share prompts with explicit permission to adapt. The goal is not for every librarian to use the exact same text, but for tested prompts to serve as starting points that colleagues modify for their own context. A norm of adapting rather than copying wholesale produces a library that improves across the team instead of converging on a single approved text.

Maintaining a living prompt library: dating, testing, and pruning

A prompt library that is not maintained degrades. AI models update on vendor schedules that are not always clearly announced. A prompt that produced reliable output on one model version may behave differently after an update: sometimes better, sometimes worse, sometimes just different enough that the library needs updating. Tasks also evolve as the librarian's workflow changes, institutional priorities shift, and the tools available to patrons change. A prompt written for a database the library no longer subscribes to, or for an instruction approach the librarian has abandoned, is not merely unused. It actively misleads any colleague who consults the shared library and acts on it.

The minimum maintenance practice is dating every prompt when it is added to the library. A date takes no ongoing effort; it simply records when the prompt was last confirmed to work as described. It lets anyone consulting the library judge whether a prompt is likely still current. A prompt tested eight months ago, in a period of significant AI tool changes, deserves more skepticism than one tested last month. Beyond the initial date, the single most useful addition is a "last tested" field, updated each time a prompt is used and confirmed to produce the expected output.

Annual review is the right cadence for pruning. Once a year, or after a significant AI model update, review the library for prompts that are no longer relevant, that reference changed tools or resources, or that newer versions have superseded. This review is not a project. It is a few hours of evaluating, deleting, and updating that keeps the library from filling with obsolete entries. For example, a library that has grown for two years without review may hold three versions of the reference email prompt, each from a different stage of the librarian's

thinking, with no sign of which is current. A pruned library holds one: the best current version, clearly dated.

The final maintenance practice is noting versions for significant revisions. When a prompt changes substantially, through a structural revision that changes what it produces rather than minor wording adjustments, note what changed and why. For example: "Revised October 2025: added constraint on response length after observing that earlier version produced outputs patron recipients found too long." Not every small adjustment needs such a note, but changes that reflect a meaningful lesson about what the prompt must include do. In shared libraries, version notes also tell colleagues why a prompt they relied on has changed, so they are not confused when it produces unexpected output. A prompt library maintained with dating, testing, and periodic pruning stays a genuinely useful professional resource, not an archaeological record of past experiments.

PRACTITIONER NOTE

A common resistance to building a formal prompt library is the perception of overhead: one more thing to maintain. What usually changes the calculation is a concrete cost. Someone spends twenty minutes trying to reconstruct a prompt that worked well three weeks earlier and cannot reproduce the output quality, because the exact phrasing is gone. The time spent trying to reconstruct it exceeds the time it would have taken to save it properly the first time. That is a specific, measurable loss, not a hypothetical future efficiency. Librarians who start saving prompts consistently after such a moment often find, six months later, that the library has forty or more entries covering most recurring tasks. The ones used weekly save time that is immediately noticeable. The ones used monthly save time that adds up across a semester. The ones saved but not yet used cost nothing except thirty seconds to capture. The ratio is favorable.

KEY TAKEAWAYS

- A prompt library solves the variability problem: the same well-constructed prompt produces consistent output that requires light editing, while reconstructed prompts produce variable output that requires substantial revision.
- The highest-value starting categories for a library prompt library are patron communication, instruction, administrative writing, and research support; begin with the five or six tasks you perform most frequently where AI has already proven useful.

- A reusable prompt has four structural components: a constant role or context specification, a task description with bracketed variables for elements that change, a constraint set specifying what to include or avoid, and an output format specification.
- Claude Projects and equivalent persistent context features (ChatGPT Custom Instructions and Custom GPTs) eliminate the need to re-establish professional context in every session; prompts used within a persistent project context produce better-grounded outputs with less per-session setup.
- Prompts shared without context are rarely adopted; share prompts with a single-sentence description of what they produce, the task they are designed for, and explicit permission to adapt for different institutional contexts.
- A prompt library that is not maintained degrades; the minimum maintenance practice is dating every prompt when added, reviewing and pruning annually, and noting significant revisions so colleagues understand why a standard prompt has changed.

References

APA 7TH EDITION

1. Dakan, R., & Feller, J. (2025). *AI fluency: Framework & foundations* [Online course]. Anthropic Academy. <https://anthropic.skilljar.com/ai-fluency-framework-foundations>

ACRL AI Competencies covered

Use & Application

Sub-competencies: 4.3, 4.1 · ACRL AI Competencies (2025)