

# Commonplace — a white paper

*The philosophy of a local-first research scrapbook, and why it is named after one of the oldest knowledge tools we have.*

---

## § 1. What a commonplace book was

Long before search engines, browser bookmarks or note-taking apps, careful readers kept a **commonplace book**: a personal volume into which they copied the passages, facts, arguments, quotations and observations they wanted to keep. The practice runs from antiquity through the Renaissance and into the eighteenth and nineteenth centuries. Students were taught to keep one; scholars, lawyers, preachers, scientists and writers lived by theirs.

A commonplace book was not a diary and not a finished work. It was a **working store of borrowed and original material**, organised so the keeper could find a passage again months or years later and put it to use — in an argument, a sermon, an essay, a decision. The word itself comes from the Latin *locus communis* and the Greek *koinos topos*: a "common place", a heading under which related material was gathered.

The decisive problem was never *collecting* — paper and ink were cheap enough. The problem was **retrieval**. A book filled front-to-back in the order things were encountered becomes useless the moment it is full: you cannot find anything. So the tradition developed techniques for organisation and indexing. The most famous is **John Locke's method** (published 1685), a clever scheme for indexing a commonplace book by the first letter and first vowel of a keyword, so that new entries could be added anywhere and still be found later.

That tension — between the freedom to gather *anything* and the discipline needed to *find it again* — is the entire design problem of this program. Commonplace is a commonplace book for the age of the web.

## § 2. Why revive the idea now

We collect more than any commonplace-keeper ever could: articles, PDFs, web pages, screenshots, half-formed notes, links we mean to read. The tools we use to collect, however, pull in opposite directions:

- **Bookmarks** save a *location*, not the thing. The page changes or vanishes and the bookmark rots. They are barely searchable and have no notes.
- **Read-it-later apps** are tuned for a queue you drain, not a collection you keep and revisit. Most are cloud services that own your material.
- **Note apps** (the OneNote / Obsidian / Notion family) are excellent for writing but treat a saved article as "content dumped onto a page" — not as a structured, searchable record with provenance.
- **Reference managers** (Zotero and kin) model academic citation beautifully, but the ceremony is too heavy for "I just want to keep this and find it later".

Each is good at part of the job. None is a *commonplace book*: a single, durable, searchable place where anything you find can live as a first-class record, with your own notes beside it, under headings that make sense to **you**.

That is the gap this program is built to fill. As the founding brief ([Initial-thoughts.md](#)) put it: *do not build a notes app — build a searchable scrapbook.*

## § 3. First principles

Commonplace is built on a small number of convictions. Everything in the product should be traceable back to one of them.

### 3.1 Every saved thing is a structured, searchable record

The historic commonplace book's superpower was that its contents were *findable*. Our equivalent is that **every item is a record, not just a blob of content**: it carries a title, a type, a source URL, an author and site, the date you captured it, a status, your notes, tags, and the full extracted text of whatever you saved.

Because of this, the same item can be found by full-text search, filtered by tag, listed under a section, or pulled up by its source — and a saved article is no longer "content on a page" but an object the system understands.

### 3.2 Retrieval is the whole point

Collecting is easy; finding is hard. So **fast, full-text search is the heart of the program, not a feature bolted on the side**. Every searchable surface — an article's text, a PDF's extracted words, a captured web page, *and your own notes* — feeds one index. Tags are a lightweight classification system, not decoration. This is Locke's indexing problem, solved with SQLite's FTS5 instead of the first-letter-first-vowel rule.

### 3.3 Your material is not trapped

A commonplace book sat on your shelf; it was *yours*. The modern equivalent is **local-first**: the database and every saved file live in a readable folder on your own disk — not inside an opaque blob, not on someone else's server. You can open the folder, back it up, copy it to another machine, or walk away from the program entirely and still have your files. Nothing about the format demands a subscription or a network connection.

### 3.4 The familiar shelf metaphor

People already know how to use a shelf of books with tabbed sections. Commonplace keeps that mental model deliberately:

*Library* → *Book* → *Section* → *Item*

A *Library* is the whole store. A *Book* is a collection on a theme (Theology, Radio, Software Ideas). *Sections* (and sub-sections) are the tabs within a book. *Items* are the things you save. The structure is visual and tactile; the search and tagging give it the retrieval power the paper version never had.

*A note on names. The user-facing collection level is a **Book**. In the code it is still the **Notebook** type, and the projects and namespaces remain **Scrapbook.\*** — a deliberate choice to rebrand the experience without churning the codebase. The mental model in the original brief was "Library → Scrapbook → Section → Item"; "Book" is simply the friendlier word for that second level, in keeping with the commonplace-book idea.*

### 3.5 Capture should be effortless; curation should be possible

Two different moments deserve two different amounts of friction. **Capturing** something — attach a PDF, capture a web page, paste a note, drop in images — should be as close to one gesture as possible. **Curating** — tagging, filing into sections, writing why you saved it, reordering, recolouring — should be *possible and pleasant* but never *required*. You should be able to hoard now and organise later, exactly as a reader once clipped first and indexed afterwards.

### 3.6 Notes belong with the thing, and in the index

A commonplace book's value was often in the keeper's *own* marginal notes, not just the copied passage. So in Commonplace your notes are first-class: they sit beside every item, and when you save, they are folded into the item's searchable text as a clearly-marked **Notes** section. A search for a word you wrote in a note finds the item — your thinking is as findable as the source.

## § 4. What an item can be

The unit of the system is the **item**. An item is one saved thing of a known type:

Type	What it is
<b>Article</b>	A read-only imported Word/RTF document, rendered in a clean view.
<b>PDF</b>	An attached PDF: text extracted for search, pages rendered for reading, with highlights and sticky notes.
<b>Link</b>	A captured web page — fetched, saved as a snapshot, and shown as a clean reader view plus extracted text.
<b>Note</b>	A Markdown note written in-app, with live preview.
<b>Images</b>	One or more images filed together.
<b>Attachment</b>	Any other file kept for the record.

Whatever the type, the item carries the same metadata spine (title, source, author, site, date, status, tags, notes, extracted text) — which is exactly what lets one search and one tag system span all of them.

## § 5. What Commonplace is *not*

Defining the edges is as important as defining the centre:

- **Not a word processor.** Articles are imported and read, not authored in-app. Editing happens in the tool that made the document; Commonplace keeps and finds.
- **Not a citation manager.** There is no bibliography engine or citation-style ceremony. It is for *keeping and revisiting*, not for formatting references.
- **Not a cloud service.** There is no account, no server, no sync-as-the-product. Sync, if it ever comes, is a convenience layered on top of files you already own.
- **Not "another notes app".** Writing is supported, but the organising idea is the *collection of saved things*, not the page of prose.

## § 6. The shape of the road ahead

The founding brief sketched a sensible progression, and it still holds. The early collection-and-retrieval core comes first; the conveniences that make capture frictionless come next:

- **Now / near term:** the Library → Book → Section → Item hierarchy; full-text search across titles, notes and extracted content; tags (now with a tree, reordering, and promote/demote); PDF reading with highlights and notes; article import; web-page capture; images and attachments; local backup and restore.
- **Later:** a browser clipper for one-gesture capture; OCR for image and scanned text; saved searches; duplicate detection; related-items / backlinks; richer export (Markdown, HTML, PDF). Sync, deliberately, last.

Each addition is judged against the principles in §3. If a feature does not make saved things easier to *keep*, *find* or *revisit* — or if it would trap the user's material — it does not belong, however fashionable it is.

## § 7. In one sentence

*Commonplace is a local-first commonplace book for the web: a single, durable, searchable place to save what you find, tag it, and revisit it — built on the oldest idea in personal knowledge work, with the retrieval power its paper ancestors never had.*