Research Source Organizer Claude Skill

Claude skill - preview and full file inventory

File inventory (everything in product.zip, inside Research-Source-Organizer-Claude-Skill/)

LICENSE.txt	996 B
README.md	2.6 KB
README.txt	2.6 KB
research-source-organizer/SKILL.md	4.8 KB
research-source-organizer/examples/claims_matrix_example.md	2.2 KB
research-source-organizer/examples/output/bibliography.md	1.9 KB
research-source-organizer/examples/output/issues.md	122 B
research-source-organizer/examples/output/sources_clean.csv	1.3 KB
research-source-organizer/reference/citation_patterns.md	1.5 KB
research-source-organizer/scripts/sources.py	9.4 KB
research-source-organizer/tests/sample_sources.csv	1.5 KB
research-source-organizer/tests/test_sources.py	1.6 KB

Tested here

PASS - dedupe, merge, APA/MLA formatting and metadata checks verified

Sample: Research-Source-Organizer-Claude-Skill/research-source-organizer/SKILL.md (excerpt 1/1)

```
---
name: research-source-organizer
description: Organises a messy pile of research sources (pasted citations, URLs, PDFs,
  reading notes) into a clean, de-duplicated source table with source-type and evidence
  tiers, theme tags mapped to the user's research questions, a claims-and-evidence matrix
  with page or section locators, a list of gaps and conflicts, and an APA 7 / MLA 9
  bibliography. Use when the user asks to organise, sort, tag, de-duplicate, tabulate or
  cite sources for a literature review, thesis, report, article or due-diligence memo.
---

# Research Source Organizer

You turn a disorganised set of sources into a research base the user can write from.
  Accuracy beats completeness: you never invent authors, years, DOIs, page numbers, quotes
  or findings. Anything you cannot confirm from what the user gave you (or from tools you
  actually used) is marked `missing` or `unverified`.

## Step 1 - Intake

Ask once (skip what the user already said):
1. The research question(s) or the section headings of the piece being written. These
  become the themes.
2. Citation style: APA 7, MLA 9, or both (default: APA 7).
3. What you have: a citation list, a CSV/Zotero export, PDFs, URLs, or notes.

Accept any mix. For each item record exactly what the user provided. If you have web or
  file tools, you may look up missing metadata and must note the source of each fill ("DOI
  from the PDF first page", "year from publisher page").

## Step 2 - Build the source table

One row per source, columns as in `tests/sample_sources.csv`:
`id, type, authors, year, title, container, volume, issue, pages, doi, url, publisher,
  accessed, themes, notes`

- `type`: article, book, chapter, report, web, preprint, news, dataset.
- `authors`: `Last, First; Last, First` or an organisation name.
- Save it as `sources.csv`. If you can run code, run the formatter, which de-duplicates
  (same DOI, or same normalised title + year), merges the duplicates' themes, flags missing
  metadata and writes the bibliography:

```bash
python3 scripts/sources.py sources.csv --out-dir out --style both
```

Outputs: `out/sources_clean.csv`, `out/bibliography.md`, `out/issues.md`. Standard library
  only.

If you cannot run code, de-duplicate by DOI and title yourself and format citations using
  the patterns in `reference/citation_patterns.md`.

Step 3 - Classify each source

Add two judgements to each row (in `notes` or extra columns):
```

SAMPLE

Preview excerpt only - the full file and all other files are in the download.