FREE LEARNING STUDIO

Build one useful app. Keep what you learn.

A practical Lovable workbook for Resource Shelf: a small library of six useful resources.



Illustrative creative scene made with Higgsfield. Not product-interface footage.

Your first complete result

A visitor can find a relevant worksheet, inspect its details, download the real file and save it in this browser. Start with public sample content. Add accounts and shared data only when the project needs them.

Inside: a worked brief, a six-record data model, focused build exercises, test evidence, a launch decision and a measurement plan.

By Ashley Kays / AI Superpower
Independent learning material | November 19, 2026

START WITH THE RESULT

A small app. A complete journey.

Use this sequence alongside the free tutorial hub. The exercises are original practice recommendations; your results come from what you build and verify.

Stage	Make	Evidence to keep
Brief	One person, one task, clear boundaries	One-page brief and acceptance checks
Build	Library, search, filter, details, files, local saves	A complete path you can demonstrate
Test	Happy path and important failure states	Expected/observed results and retest notes
Publish	The actual hosted URL and its files	Signed-out release check and recovery plan
Learn	A small distribution and measurement test	Dated observations and a next decision

Before you open the builder

- [ ] Open the six Markdown worksheets and inspect their CSV records.
- [ ] Prepare fictional practice data and decide what is deliberately excluded.
- [ ] Check your workspace credits, current plan and any connected-service costs.
- [ ] Keep a folder for the brief, screenshots, test records and release notes.

A useful boundary

This workbook does not establish that the example was built in a signed-in Lovable account. The teaching demo shows intended behavior. Run the exercises in your own workspace and record what actually happened.

References: L1: platform overview | L13: credits and usage

SCOPE THE FIRST RELEASE

Write a brief you can test.

The starter example is filled in below. Use Lovable-App-Brief.md to make an editable copy for your own idea.

Decision	Resource Shelf example
Audience	A freelance designer who wants useful templates in one place.
Core job	Find, inspect and download a relevant resource; optionally save it.
First screens	Public library, resource details and saved-only view.
Data	Six public fictional resource records and local saved IDs.
Included	Title/summary search, topic filter, result count, reset and real file links.
Excluded	Accounts, uploads, payments, private client documents and external AI calls.

Five checks that define the result

- Searching portfolio finds its outline; uppercase and padding behave the same.
- Selecting Research shows the interview planner; conflicting filters can be reset.
- A known detail route opens directly; an unknown ID has a useful not-found state.
- Each available download opens the correct real worksheet.
- Saved items survive refresh where storage works, with the local-only boundary explained.

My one user, one task and clearest success check

References: L2: Plan mode

PREPARE THE CONTENT

Six records. Real files.

Inspect the data before asking the builder to render it. A good dataset makes the desired behavior concrete.

Resource	Topic	Useful outcome
Client Kickoff Checklist	Client work	Prepare the goal, scope, decision owner, materials and next steps for a client kickoff conversation.
Portfolio Case Study Outline	Portfolio	Explain the problem, your contribution, evidence, decisions and outcomes of one design project.
Research Interview Planner	Research	Plan neutral questions about a recent real experience and separate observations from interpretations.
Landing Page Review Sheet	Web design	Review one landing page for audience clarity, promise, proof, action and mobile usability.
Accessibility Spot Check	Accessibility	Inspect keyboard use, labels, headings, contrast and error feedback, then record issues for follow-up.
Weekly Design Practice Planner	Practice	Choose one design skill, plan short practice sessions, record feedback and set the next exercise.

Keep a simple data contract

id stays stable when a title changes. **title** and **summary** are searched. **topic** supports filtering. **format** and **level** help a visitor choose. **download_url** must point to an actual file. **updated_date** records the content revision.

Try a controlled variation

In a copy of the data, make one title longer and one download URL blank. The layout should wrap the title and explain that the file is unavailable. Restore the approved manifest before publishing.

Which field or state exposed an assumption in my design?

WORK DELIBERATELY

Use the mode that fits the task.

Current documentation distinguishes Chat, Plan and Build. Older material may call Build “Agent mode” and Instructions “Knowledge”.

Mode	Use it for	A useful request
Chat	Explore a decision	Does this first version need accounts? Explain the tradeoff.
Plan	Review the approach before changes	Describe search, filtering, states and checks. Do not implement yet.
Build	Implement an agreed change	Build the approved search behavior and verify the named cases.

Put stable decisions in Instructions

A starting instruction to adapt

Resource Shelf is a public learning library for freelance designers. Version one uses six approved records, real files and browser-local favourites. Preserve the static-first scope, white-and-green design, semantic controls and clear empty/error states. Never invent file availability, customer results or verification. Ask about a material ambiguity and report what was actually tested.

Keep credentials and temporary tasks out of persistent Instructions. Review generated changes against the brief; an instruction is guidance, not proof that every result follows it.

One project rule I want to keep stable

References: L2: Plan mode | L3: Build mode | L4: Instructions

BUILD THE FIRST USEFUL SLICE

Make the whole task work.

A complete slice connects the controls, the content and the outcome. Check the result before asking for the next feature.

Library

Render the six approved records with readable titles, summaries and topics.

Search and filter

Match title and summary without case sensitivity. Trim padding. Combine search and topic using AND logic.

Details

Use stable IDs for routes. Direct visits and refresh must show the correct record.

Download

Use the real manifest URL. Missing files get an unavailable state, not a guessed link.

Focused build request

Implement search across title and summary, an accurate result count and Clear filters. Keep the six-record dataset. Verify portfolio, PORTFOLIO, a padded query, a no-match query, a conflicting topic and reset. Preserve the existing design and report the cases actually checked.

What did I observe in the first complete journey?

References: L3: Build mode | L6: testing

FINISH THE USEFUL DETAILS

Make saving and delivery honest.

The first saved list belongs to this browser. It is not an account, a backup or cross-device sync.

Situation	Expected behavior
Save and refresh	The same stable resource ID remains saved when local storage is available.
Remove and refresh	The removed ID stays removed; an empty saved list explains what to do.
Blocked or invalid storage	The app stays usable and explains any persistence limitation.
Old saved ID	Ignore missing resources safely rather than crashing or opening another item.
Real download	The file opens and matches its title, format and description.
Missing download	Show an honest unavailable state and a useful return action.

Inspect the whole file experience

- [ ] Open every worksheet, not just the ZIP archive.
- [ ] Check the download control with a keyboard and on a narrow screen.
- [ ] Keep a copy of the manifest that belongs to the published release.
- [ ] Measure a download click as a click; do not label it completed learning.

File or persistence issue / fix / result after repeating the check

References: L6: testing

IMPROVE THE EXPERIENCE

Design for the real states.

The default screen is one state. Good practice includes long text, empty results, keyboard use and a small viewport.

Review area	What to try	What to record
Reading	Long title, enlarged text, short and long summaries	Clipping, hierarchy and whether the purpose stays clear
Keyboard	Search, filter, details, save, download and return	Focus visibility, order, activation and any trap
Meaning	Labels, saved state, result counts and errors	Whether words explain state without relying only on color
Responsive layout	390px, 768px and 1440px; inspect all important states	Widths and states actually checked; overflow or hidden controls

Give targeted visual feedback

Name the element and the user task: “Keep the topic label visible, let the selected topic wrap, and stack the controls at narrow widths.” Use the current preview toolbar to target changes. Check current usage allowances; do not assume every visual action is free.

Use Accessibility Spot Check as an early review aid. It does not establish full standards conformance. Record manual checks, measured contrast and the need for any specialist review separately.

The highest-impact usability issue I found

References: L5: preview toolbar | L6: testing | L13: credits and usage

DEBUG WITH EVIDENCE

A defect needs a repeatable story.

The test matrix contains 26 proposed cases. Its observed-result and status columns are intentionally blank.

Case	Expected result	Observed result
Search portfolio	Only the portfolio outline appears	Record after running
Portfolio + Research	Zero results, count 0 and useful reset	Record after running
Unknown resource ID	Not-found state, no unrelated content	Record after running
Save, then refresh	Saved item persists where storage works	Record after running

A useful defect report

Copy this structure

Environment: preview or published URL, browser and width.
Action: the exact steps and input.
Expected: the behavior in the brief.
Observed: what happened, including a redacted error.
Evidence: screenshot, console detail or test output.
Retest: repeat the same case after the smallest relevant fix.

Use existing test tools for a confirmed regression risk. Keep a known working code version. A code restore is not a database rollback; schema, stored data and configuration may need separate recovery.

One failed check and the evidence that changed my diagnosis

References: L6: testing | L15: version history

OPTIONAL EXTENSION

Add shared data deliberately.

Move beyond the static version only for a clear benefit, such as account-based favourites across devices or controlled catalogue editing.

Data or action	Intended access
Published resource records	Public read; approved administrators manage changes.
Account saved-resource links	Only the signed-in owner can read, create or delete their saved links.
Private provider credentials	Trusted server work only; never shipped in browser code or screenshots.

Lovable offers a built-in backend; an external Supabase project is another option. Record the connected service, billing owner and migration plan. Do not assume the options are interchangeable with a one-click move.

Permission evidence to keep

- [ ] With synthetic accounts A and B, test that B cannot read or delete A’s saved record through a direct request.
- [ ] Test that an ordinary user cannot change the public catalogue.
- [ ] Verify ownership in database rules and privileged server functions, not just hidden controls.
- [ ] Keep public browsing open and explain any proposed upload of local saves.

Pause before a paid integration

Name the user benefit, data sent, permission, cost, private credential location, server checks, timeout and failure behavior. Use the secure secret mechanism; never paste privileged keys into an ordinary prompt or public file.

References: L9: built-in backend | L10: authentication | L11: database access | L12: secrets

PUBLISH AND HAND OVER

Verify the URL people will use.

A working preview is useful; a tested release requires checking the actual published site and its files.

- [ ] Keep a known working version and record the release identifier.
- [ ] Document the repository, installation/build commands and environment variable names without secret values.
- [ ] Use the intended publishing workflow and inspect its security findings.
- [ ] Open the returned live URL in a signed-out context and repeat the main task.
- [ ] Directly open a detail URL, refresh it, and inspect a missing route.
- [ ] Check all six worksheet downloads and the complete ZIP.
- [ ] Record domain, hosting, service billing and maintenance ownership.
- [ ] Write recovery steps for code and any separate data/configuration changes.

Code portability is a separate check

Git sync makes code available in the connected workflow; running the app elsewhere still depends on its framework, build, environment and external services. If an independent build was not executed, label it unverified.

Release URL / version / owner / remaining launch issue

References: L7: GitHub sync | L8: publishing | L15: version history

SEARCH AND AI VISIBILITY

Make the content useful and reachable.

SEO, answer visibility and a good learning experience share practical foundations: accessible pages, original detail and trustworthy evidence.

Page element	Resource Shelf starting point
Title	Resource Shelf: practical worksheets for freelance designers
Description	Find and download project, portfolio, research and review worksheets.
Main content	Clear titles, real descriptions, working file links and meaningful topic pages.
Identity	Accurate author/publisher, revision date and transparent teaching-example label.
Technical checks	Canonical URL, crawl access, direct-route behavior, internal links and sitemap.
Rich media	Helpful visuals with descriptions; transcripts for actual instructional video.

Use structured data only when it matches visible content. Do not invent reviews or add schema as a promise of search treatment. Google’s current guidance does not require special AI markup or llms.txt for visibility in its AI search features.

Measure the right surface

Search Console can report search performance and supported generative-AI impressions. Check the site’s AI inclusion setting and indexing status. Bing AI Performance reports citations across supported experiences. Neither report is a count of all AI-driven site visits.

The search question my page answers, with a specific useful example

References: L14: SEO and AI search | G1: AI search guidance | G2: AI performance report | G3: AI inclusion control

DISTRIBUTION AND MEASUREMENT

Run a small learning experiment.

Choose two channels where the audience has a reason to use the resource. A listing is an asset; traffic is an observed result.

What you learn	Measure	Do not confuse it with
Did people see it?	Search/AI impressions or supported citation reports	Visits to your site
Did they arrive?	Page views, users, sessions and source	Unique people inferred from view counts
Did they use it?	Lesson views, prompt copies, exports, download clicks	Completed reading or proven skill
Did business follow?	Qualified inquiries, paid orders and collected revenue	Free joins or promised earnings

A first two-week test

- Days 1-2: test delivery, privacy controls and fixed analytics events.
- Days 3-5: share one useful native lesson and one permitted community contribution.
- Days 6-9: answer real questions and improve the confusing step.
- Days 10-14: compare useful actions and effort; continue, change or pause.

Whop is an optional delivery channel

Create a Free product, attach the intended Files/Courses access, complete the storefront and inspect Discover requirements. Test a fresh member’s access. Keep tracking links separate by placement. The bonus listing draft is prepared copy, not a live product.

Use the blank Launch Tracker: one metric per row, with period, unit, report source and limitations. Leave unknown values blank. Full channel rules and Monetise evidence distinctions are in DISTRIBUTION-PLAN.md.

References: W1: storefront and products | G2: AI performance report | B1: AI Performance

YOUR RELEASE DECISION

Keep a record worth revisiting.

Fill this after testing. A remaining gap is useful information; it should not become a hidden assumption.

Project / user / one useful task

Published or preview URL / release version / review date

Area	Evidence or remaining issue	Status
Main task and files		
Keyboard and mobile		
Data or permission boundary		
Published routes and recovery		
Measurement and ownership		

Decision: share / improve first / stop. Why?

One next experiment / owner / review date

KEEP LEARNING

Sources, files and reuse.

Platform facts link to primary documentation. Exercises, prompts and sample worksheets are original AI Superpower material.

L1 Lovable: platform overview
L2 Lovable: Plan mode
L3 Lovable: Build mode
L4 Lovable: Instructions
L5 Lovable: preview toolbar
L6 Lovable: testing
L7 Lovable: GitHub sync
L8 Lovable: publishing
L9 Lovable: built-in backend
L10 Lovable: authentication
L11 Lovable: database access
L12 Lovable: secrets
L13 Lovable: credits and usage
L14 Lovable: SEO and AI search
L15 Lovable: version history
G1 Google: AI search guidance
G2 Google: AI performance report
G3 Google: AI inclusion control
B1 Bing: AI Performance
W1 Whop: storefront and products

Make these resources yours

Use and adapt the original prompts, worksheets and sample data for personal, teaching and client projects. If you redistribute the unchanged free kit, keep AI Superpower’s credit, the tutorial link, START-HERE.md and the source register. Do not imply endorsement by Lovable or another platform.

The kit includes editable Markdown, CSV data and tests, six actual sample worksheets, a launch checklist and distribution notes. Full exact source URLs, current-version notes and evidence boundaries are in SOURCE-REGISTER.md.

Continue: https://aisuperpower.cc/learn/lovable
Practice session: aiteachers.cc/learn/lovable-practice/