

Turn a page into a working webapp.

Build a checklist that adds, completes and removes tasks.

Build one thing

Check it works

Choose an extension

Before you start.

01

You already know

Module 03. Keep a backup of working HTML.

02

Open these tools

AI chat, editor, browser; localhost server for reliable storage tests.

03

Check the requirement

Local JavaScript needs no API or backend. Storage stays in this browser.

Your first small win.

Create a complete single-file HTML task checklist. Add a task, mark it done and delete it. Use crimson, ink and near-white. Start empty. No network requests, accounts or libraries. Include labels and keyboard controls. Return code only.

[Open copyable prompt](#)

What happens between input and output?

- 1 An input changes application state.
- 2 JavaScript updates the view and handles actions.
- 3 Local storage is not shared or secure cloud storage.

Step 01. Start a new app folder.

- 1 Create my-checklist-app.
- 2 Generate the quick-win code.
- 3 Save it as index.html and open it.

Step 02. Try three actions.

- 1 Add the fictional task Buy pens.
- 2 Mark it complete.
- 3 Delete it and check the list is empty.

Step 03. Add persistence.

- 1 Use the build prompt for localStorage.
- 2 Save a backup before replacing code.
- 3 Ask for an Export JSON and Clear data control.

Step 04. Serve it locally.

- 1 With Python installed, run `python3 -m http.server 8000` in the folder.
- 2 On Windows, `py -m http.server 8000` is an alternative.
- 3 Open `http://localhost:8000` rather than the file URL.

Step 05. Test state and reload.

- 1 Add two tasks and reload.
- 2 Export the list, then clear it.
- 3 Confirm the cleared state remains after another reload.

Step 06. Keep the scope small.

- 1 Use fictional task text.
- 2 Do not add API keys to browser code.
- 3 Explain that another browser has a different local list.

Make the working version.

Add localStorage persistence, Export JSON and Clear data to my checklist. Treat typed text as text, not HTML. Handle empty inputs and damaged stored JSON. No API keys or network calls. Explain how to test on localhost. Code: [paste].

[Open copyable prompt](#)

Ask for a review. Then check it.

Review this checklist code for unsafe HTML insertion, missing labels, empty input handling and persistence failures. Give manual tests, not claimed results. Keep the app offline and single-file. Code: [paste].

[Open copyable prompt](#)

Prove your result works.

- 1 Add, complete and delete work.
- 2 Reload keeps state on localhost.
- 3 Export and clear work; typed markup is displayed as text.

When it does not work yet.

01

Storage differs by URL

Use the same localhost address and port.

02

A revision broke it

Restore the backup, then change one feature.

Choose your next challenge.

01

Core complete

Show one checked output to a partner.

02

Extension 1

Add filtering and an undo action.

03

Extension 2

Build a different tool with the same state pattern.

Keep the result. Keep the evidence.

Build a checklist that adds, completes and removes tasks.

Save your files

Save your prompt

Name the next change

Sources you can check.

01

MDN

localStorage

02

MDN

Your first website