

Neighbourhood Swap: your first useful website

Build a small site someone could use to find a repair session or swap event.

Build

Check

Try independently

The situation and your goal

Situation

The fictional neighbourhood has three events. Residents need to see what, when and where, then use a clear contact link. You will build a real browser page from an approved data sheet.

Keep this artifact

A three-event website, a written brief and a browser test checklist.

Understand the mechanism

01

HTML carries meaning

Headings, lists, navigation and buttons tell browsers and assistive tools what content does. A visual div is not automatically an accessible control.

02

CSS controls presentation

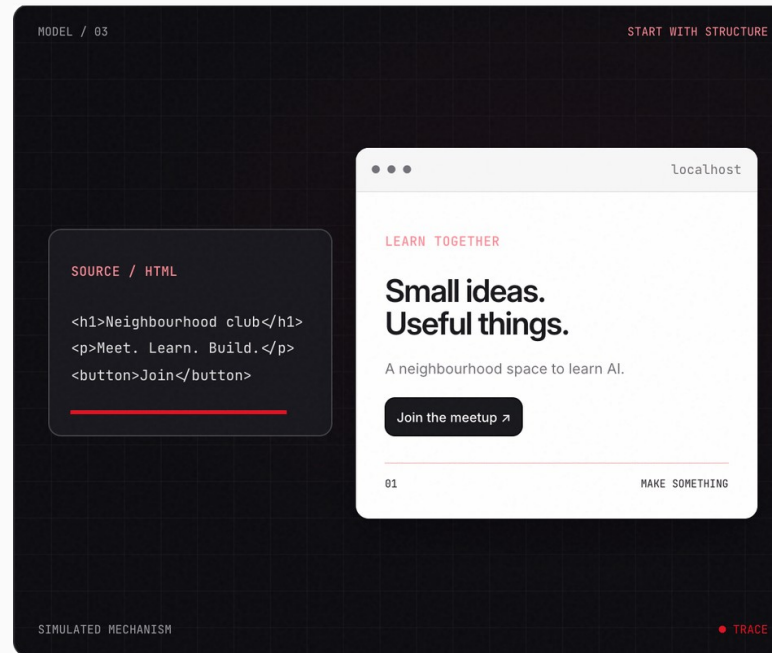
Spacing and responsive rules should adapt the same content to different screens. Content should remain readable when the viewport changes or text is enlarged.

03

A website is an artifact

A prompt is a request for a file. The deliverable is the saved code that opens in a browser and survives your tests, not a screenshot of a chat answer.

Explore before you build



Define the visitor job

1

Read swap-events.csv and choose one visitor question: which event can I attend this weekend? Write the required information and one primary action.

2

Check: Your brief names the visitor and an observable success action.

3

Hint: Use only the fictional supplied events.

Build the semantic page

- 1 Ask for one index.html with header, main, three event cards and footer. Save the code as UTF-8 and open it in a browser.
- 2 Check: All three records appear without invented details.
- 3 Hint: Check that the file extension is .html rather than .html.txt.

Inspect the code

- 1 Find the h1, event headings, links and media query. Explain which section contains content and which contains styling.
- 2 Check: You can edit one event title without asking AI.
- 3 Hint: Use browser developer tools to inspect the rendered element.

Improve the small screen

- 1 Use a single-column layout on narrow screens, visible focus styles and comfortable tap targets. Test at 320px and 390px.
- 2 Check: No horizontal overflow or clipped event title appears.
- 3 Hint: Do not hide overflow to conceal content that is too wide.

Test real actions

- 1 Use keyboard Tab to reach the contact link, check its destination and test the page with a long event name.
- 2 Check: The action works and focus remains visible.
- 3 Hint: A button with no behavior is not a working signup.

Document and hand off

- 1 Save a screenshot, the source file and a short note about what the page does and does not do.
- 2 Check: A partner can open the file and identify the primary action.
- 3 Hint: This first version has no backend or real booking system.

When it goes sideways

01

The browser shows code as text

The file is saved with the wrong extension or contains markdown fences. Try: Save only the HTML content as index.html.

02

It looks good but the link does nothing

The generated page uses a placeholder destination. Try: Replace the placeholder with a real approved contact link or clearly label the demo.

03

Mobile needs sideways scrolling

A fixed-width element exceeds the viewport. Try: Use responsive widths and inspect the actual overflowing element.

Review your evidence

1

Three approved events are displayed accurately.

2

Semantic headings and a main region are present.

3

Keyboard focus and contact link work.

Review your evidence

1

320px and 390px layouts stay readable.

2

You can change content directly in the source.

What is the real deliverable?

- 1 A screenshot of an AI answer.
- 2 The saved page and evidence that it works.
- 3 A longer prompt.

Which change belongs mainly to CSS?

- 1 Event date.
- 2 Meaning of the page heading.
- 3 Responsive spacing between cards.

Build with a clear contract

Build a single-file responsive Neighbourhood Swap website from this fictional dataset: Repair Cafe, Saturday 10:00, community room; Book Swap, Saturday 12:00, reading corner; Seed Exchange, Sunday 11:00, covered courtyard. Use semantic HTML, clear event cards, keyboard-visible focus and a narrow-screen single-column layout. No fake signup backend. Explain where I edit content and how I test the contact action.

[Open copyable prompt](#)

Create a repair-cafe microsite

- 1 All new facts come from your approved brief.
- 2 FAQ answers unknown questions honestly.
- 3 A partner completes the primary action.

Build independently, then teach it back

Your next challenge

Change the visitor job to finding a repair session. Add a FAQ and one accessibility improvement without copying the original prompt.

Explain your decisions

Show your artifact, one test that passed, one failure you repaired and one boundary the system respects.