

NGP SHED

A LITTLE LESS TO REMEMBER.

A digital shed for Nottz Garden Project: garden jobs,
loose thoughts and the useful things in between.



Actual NGP poster reference.
Not a screenshot or live timetable.

What works today. What could grow next. How to put the
project on your website.

INSIDE THE GUIDE

02-04	Purpose, Home and garden notebooks
05-06	Onboarding, design and architecture
07-08	Future potential and next milestones
09-10	Website options and publishing notes

Prepared 3 October 2026 / Working prototype, not a finished service

01 / WHY THE SHED EXISTS

The Shed starts with a simple idea: useful thoughts need somewhere easy to land.

Community gardening creates small but important things to remember: bring gloves, order compost, call someone back, keep a receipt, or save an idea for next summer. NGP SHED brings those fragments and garden notebooks into one browser app.

Its intended users are the people organising and helping with the gardens. The aim is to reduce the effort of remembering and filing things, while keeping decisions with people. That benefit is a design goal to test with NGP, not a measured outcome yet.

Built now	What it does
Today / Home	Shows sample sessions, a few reminders and recent captures.
Chuck it in the Shed	Saves a thought immediately; type and garden can be chosen later.
Gardens + details	Four garden notebooks with sample sessions, tickable jobs and linked captures.
Take tour	A five-step introduction with safe practice activities.
Local saving + motion	Notes and job ticks persist in this browser when storage works. GSAP gives short interaction feedback.

PROTOTYPE BOUNDARY

Calendar, Impact, accounts, shared team data and cross-device synchronisation are future possibilities. The sessions, reminders and garden activity shown today are demonstration content.

The useful distinction

This prototype proves an interaction and visual direction: catch a thought quickly, find a garden easily, and make the interface feel like NGP's own digital Shed. It does not yet provide a live operational record for the whole project.

02 / TODAY AND QUICK CAPTURE

Home is the place to get your bearings and catch the thought before it wanders off.

- What is happening.** Two sample session cards show a time and garden. Opening a card takes you to that garden notebook.
- Needs a look.** A short set of sample reminders can be ticked. Shared job IDs connect those ticks to the corresponding garden notebook.
- Recently chucked in.** The latest four captures are shown first. Once there are more than four, See all reveals the rest.
- The noticeboard.** An actual NGP poster sits alongside upcoming sample dates. The poster is clearly reference material, not today's timetable.

Capture first, organise later

Type a job, idea, note or receipt reminder into the yellow capture panel and submit with Enter or Chuck it in. The app trims the text, saves the item and places it at the top of the capture list. Categorisation is optional.

Action	What happens
Write + chuck it in	The item begins as Unsorted and receives an ID and creation timestamp.
Open Sort	Choose Unsorted, Task, Note, Idea, Event or Expense.
Choose a garden	Link the item to a garden, or leave it without one.
Refresh	Saved captures and categories are restored from localStorage, if the same browser origin is used.

AN IMPORTANT DIFFERENCE

A capture labelled Task is currently a categorised note. It cannot yet be completed with a checkbox, edited or archived. The tickable jobs elsewhere are predefined sample jobs.

03 / THE GARDEN NOTEBOOKS

Each garden has a place to keep its jobs, sessions and loose notes together.

Gardens opens a set of four large cards. Each card shows the garden name, location, sample next session and the number of predefined jobs still unticked. The full card is a link, which keeps navigation simple on a phone.

Notebook	Prototype content
Boulevard Gardens	Hyson Green. Sample Saturday session, jobs and an actual Boulevard poster.
Gamble St Corner	Radford. Sample Tuesday session, water-butt job and the Sally reminder.
Berridge Curve	Sample Saturday session, path clearing and autumn planting jobs.
Bestwood Community Garden	Sample Saturday session, photos and compost jobs, plus an actual NGP poster.

Inside a notebook

The detail screen has a sample next session, a Jobs checklist and its own capture panel. A note added there is linked to that garden automatically. The garden's Shed lists linked captures, and a quieter supporting column contains upcoming sample information, garden notes and activity.

How garden sorting behaves

Use Sort to change a capture's type or garden. If it is moved to another garden, it disappears from the current garden's list and is available in the destination notebook. Home remains the mixed list across gardens.

A TYPICAL USE

Open Bestwood, tick the sample compost job, and chuck in "Bring spare gloves next Saturday". The tick and captured note survive refresh when browser storage is available. The note remains separate from the predefined checklist.

All names and supplied posters are part of the prototype's NGP context. Displayed schedules, jobs, counts, quotations and activity have not been verified as current operational information.

04 / A FAMILIAR FEEL, WITH A LITTLE LIFE

The design borrows NGP's handmade character instead of turning it into a conventional productivity dashboard.

Visual direction

Black ink, plain paper and bright yellow lead the interface. Muted green and lilac support the main actions. These colours are approximations guided by the supplied posters, not official brand specifications. Chunky lettering, crooked paper edges, small sticker accents and actual poster images keep the connection to NGP visible.

Clearer hierarchy

The latest refinement gives capture the strongest colour, keeps supporting session cards quieter, shortens repeated explanations and removes inactive future-navigation labels. Garden notes and activity share one panel. Dropdowns use bold yellow-backed chevrons; Take tour has a small paper-tag treatment.

Tour step	What a person learns
1 / Come in	The Shed is a place for garden jobs and loose thoughts.
2 / Today	Explore examples of sessions, jobs and saved notes.
3 / Chuck it in	Practise saving and sorting a note, including an optional garden.
4 / Gardens	Try ticking a practice garden job.
5 / All set	Understand browser-only saving and open Home or Gardens.

Practice notes and ticks stay inside the tour. They do not change real records. Skip and Escape close it; a separate saved preference remembers that it was completed or skipped. Take tour lets people replay it.

Motion helps explain the action

GSAP and Flip provide short list insertions, unfolding Sort panels, garden-card entrances and tour paper/sticker feedback. State changes do not wait for animation. Reduced-motion preferences skip movement, and controls remain usable when GSAP is unavailable.

05 / HOW THE APP IS BUILT

It is a small static web app. The current interface runs in the browser, without an application backend.

Layer	Role
HTML + CSS	index.html loads the app. Shared styles, onboarding styles and refinements control layout, colour and responsive behaviour.
Views + sample data	home.js, gardens.js and layout.js render screens. data.js holds the illustrative content and poster references.
Interactions	app.js handles hash navigation, captures, sorting and job ticks. Onboarding has separate content, view and state modules.
Local state	store.js keeps captures and completion flags in memory and writes them to localStorage.
Optional animation	motion.js wraps GSAP core and Flip, supplied as local vendor files.
Development server	server.mjs serves local files at port 5173. It is a preview server, not a shared-data API.

Two separate saved records

The key **ngp-shed-v1** stores captures and completed job IDs. The key **ngp-shed-tour-v1** stores the tour preference. There are no accounts, database, cloud backups or team synchronisation.

WHERE A NOTE ACTUALLY LIVES

localStorage belongs to a browser origin: its scheme, host and port. Notes at localhost do not automatically appear on a public domain, another browser or another device. Clearing site data can remove them. Private browsing and blocked storage can affect retention. [2]

Normal app actions do not send capture records to an application server. Optional Google Fonts are fetched externally when available; system fonts provide a fallback. Text is escaped before it enters rendered capture markup. These choices are useful foundations, but are not a security or accessibility certification.

06 / WHAT COULD GROW NEXT

Grow from everyday use. Add the features that solve recurring problems for the people doing the gardening.

First: make the small actions dependable

Protect unfinished drafts, let people edit and archive notes, add Undo, allow captured Tasks to be completed, show each note's garden, and make failed-saving status persistent. Add export/import so local records have a practical route out of one browser.

Then: connect the garden work

A Calendar could bring sessions, seasonal tasks and deadlines together. Useful features would include simple date assignment, a weekly view and calendar-file export. Recurring jobs could cover watering, compost checks and seasonal planting. These need genuine NGP schedules and ownership, not more demo dates.

Then: make it a shared Shed

If the team needs shared information, introduce a backend, authenticated access, roles and server-side permissions. Decide who can see or change each garden's records. Add history, recovery, backups and sensible handling when two people edit the same item.

Impact as a record of real activity

An Impact area could collect attendance, volunteer hours, planting, harvests, expenses and session notes. Keep definitions consistent, allow corrections and use verified entries for summaries. It could help people describe the work to partners and funders without inventing outcomes or treating attendance alone as proof of impact.

More accessible use in the gardens

An installable app, tested offline behaviour, clearer phone layouts, readable text, photo notes and voice-assisted capture could be valuable. Photo and audio features would need deliberate consent and access choices. None of these capabilities is implemented yet.

HOW TO DECIDE WHAT COMES NEXT

Ask: does this save a real person time, keep a useful record, or prevent something being forgotten? Trial one change at a time with gardeners and coordinators. Keep capture quick and preserve NGP's voice.

07 / A SENSIBLE NEXT MILESTONE

The next release should make the existing experience more trustworthy before adding whole new areas.

Priority	Outcome to aim for
1 / Capture lifecycle	Draft recovery, editing, archiving with Undo, and completion for captured Tasks.
2 / Clear saving	Accurate per-item save state, persistent storage-failure feedback and a basic export/import route.
3 / Garden orientation	Garden labels on Home notes, clear movement feedback and predictable quick-add context.
4 / Device testing	Check phones, tablets, keyboard navigation, zoom, focus, native selects and tour scrolling.
5 / Real pilot content	Agree real session information and which jobs belong in a shared list before the prototype is used operationally.

What has been checked

Syntax checks and all 13 automated tests passed during the latest refinement. They cover saving, safe text rendering, garden state, animation fallbacks, reduced motion and tour practice. They do not verify every visual layout.

What is still a known limitation

Drafts are lost on navigation. Notes cannot yet be edited or archived; Task labels are not tickable jobs. Failed writes leave notes in memory, but their warning does not persist reliably across screens. No shared operational data is connected.

The latest styling still needs an in-browser check because inspection was unavailable. Test real phones for keyboard behaviour, tour scrolling and touch targets.

A USEFUL PILOT SESSION

Ask a few NGP users to capture a thought, find a garden, tick a sample job and explain where their note is saved. Observe where they hesitate. Use that evidence to choose the next small release.

08 / YOUR PORTFOLIO

NGP SHED can follow the same case-study and interactive-showcase pattern already used on your custom site.

I inspected **Portfolio 2026**. It is a static HTML/CSS/JavaScript site, with project-*.html case studies, a selected-work list in index.html, and local demos under showcases/. Kendamil and MuscleFood already demonstrate this approach. No new website framework is needed.

Proposed location	What belongs there
project-ngp-shed.html	A case study using your existing project-page layout and chapter navigation.
showcases/ngp-shed/	A portable copy of the working app: index.html, src/ and public/.
assets/documents/	This PDF, renamed ngp-shed-guide.pdf for a simple public link.
assets/images/	A web-ready preview image for the work list and case study.

Tell the story before opening the tool

Present the need, NGP poster references, the capture interaction, the garden notebooks and what you learned through refinement. Label the work as a prototype. Include your actual role and contribution without presenting untested benefits as achieved impact.

Give the visitor two simple actions

Use **Try NGP SHED** to open the app in its own window, and **Download the app guide** for this PDF. A full-window demo is the cleanest initial choice for phones. An optional click-to-load iframe can follow your existing Kendamil showcase, keeping NGP styles separate from portfolio styles. [3]

ONE EXISTING SCRIPT NEEDS ATTENTION

assets/js/site.js changes an unrecognised case study's secondary hero link to Discuss this project. Add an explicit NGP branch for the demo link, or place the demo action outside that selector. Also add NGP to projectDirectory so related-project navigation recognises it.

Add a selected-work entry in index.html using the current work-item structure, with a preview and the project link. This is an integration plan; the portfolio files have not been changed or published.

09 / INTEGRATION CHECKLIST

The key preparation is making the copied app work from `showcases/ngp-shed/` rather than assuming it lives at the website root.

1 / Copy the web files. Place `index.html`, `src/` and `public/` in `showcases/ngp-shed/`. Keep the local GSAP files and NOTICE. Leave `node_modules`, the development server and private working files out of the copy.

2 / Adapt asset URLs. In the copied `index.html`, change `/src/` and `/public/` references to `./src/` and `./public/`. In copied `data.js`, use `./public/` for poster paths. In copied `src/refinements.css`, the chevron URL becomes `../public/icons/select-chevron.svg`. Relative module imports already fit this layout.

3 / Connect the case study. Add the work-list card, case study, preview and PDF link. Register NGP in `assets/js/site.js` and handle its secondary hero button so the demo link is not overwritten.

4 / Verify before publishing. Run the portfolio's python3 `scripts/check_site.py`, then test the nested app separately: routes, arrows, posters, tour, capture and refresh. Check a phone and reduced motion. The portfolio checker alone does not prove the whole app works.

5 / Use the current deployment route. The portfolio README describes Netlify with no build command and publish directory `.`, plus GitHub Pages from the repository root. The live provider has not been confirmed. Both can host static web assets. [4,5]

Links from project-ngp-shed.html

```
<a href="showcases/ngp-shed/index.html">Try NGP SHED</a>
```

```
<a href="assets/documents/ngp-shed-guide.pdf">
```

```
Download the app guide</a>
```

Publishing on a new origin starts separate browser storage. Hosting the demo does not add team accounts or synchronisation. A separately hosted demo is also possible if you prefer not to copy it into the portfolio; Cloudflare Pages supports Direct Upload. [1,2]

Sources and evidence

[1] [Cloudflare Pages / Direct Upload](#)

[2] [MDN / Window.localStorage](#)

[3] [MDN / iframe element](#)

[4] [GitHub / What is GitHub Pages?](#)

[5] [Netlify / Build configuration](#)

Local evidence: NGP README, `src/`, supplied posters and UI review; portfolio README, `index.html`, `project-template.html`, Kendamil showcase, `assets/js/site.js`, `netlify.toml` and `scripts/check_site.py`. Reviewed 3 October 2026. Future features are proposals.