

Pocket BorrowBox: put your app on a phone

Install your own useful webapp and understand exactly what works offline.

Build

Check

Try independently

The situation and your goal

Situation

BorrowBox works in a browser.
You want a phone home-screen entry with a proper name, icon and an honest offline experience.

Keep this artifact

A PWA folder, an install test and an offline capability table.

Understand the mechanism

01

A PWA is a web app

A manifest supplies app identity and install information. Browser support and installation UX vary, so test the device rather than assuming every phone shows the same prompt.

02

Cache is not data sync

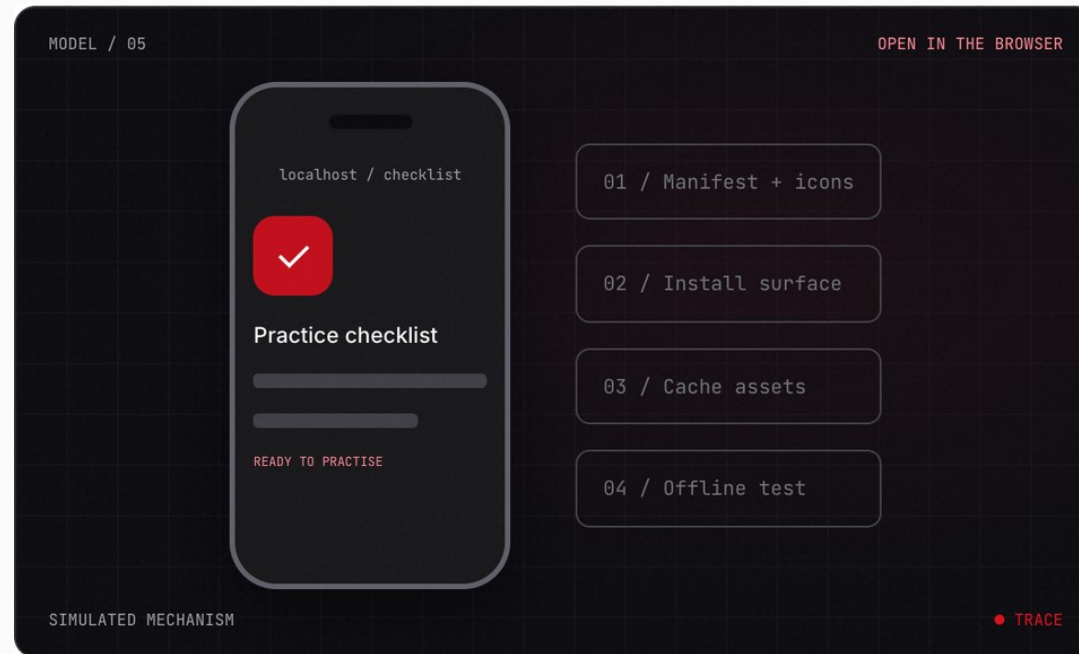
A service worker can cache static app files. It does not automatically synchronize records, run a remote AI model offline or turn localStorage into cloud storage.

03

Versions matter

An old cached file can keep showing an outdated app. Name cache versions, remove obsolete caches in activation and test an update before sharing.

Explore before you build



Prepare a real folder

- 1 Start from BorrowBox. Keep index.html, manifest.webmanifest, sw.js and the supplied 192/512 icons together.
- 2 Check: All relative paths resolve when served.
- 3 Hint: Local file URLs are not the reliable service-worker test environment.

Define app identity

- 1 Set the name, short_name, start_url, display and icons in the manifest. Link it from HTML.
- 2 Check: Developer tools recognize the manifest and both icon sizes.
- 3 Hint: Use the supplied PNG icons; do not rename an SVG as a PNG.

Serve securely

- 1 Use localhost during development. Use HTTPS for phone deployment, then open the published URL on the phone.
- 2 Check: The service worker registers without a security error.
- 3 Hint: A localhost address on your laptop is not your phone localhost.

Choose offline behavior

- 1 Cache the local static shell and approved starter files. Explain that external links and remote model calls still require connectivity.
- 2 Check: The capability table separates cached UI from network functions.
- 3 Hint: Cache only known app resources, not arbitrary sensitive responses.

Test install and offline

- 1 Use the supported install or Add to Home Screen action. Disconnect network after a successful load, then reload and update a local item.
- 2 Check: The cached app opens and its local data still works where supported.
- 3 Hint: Offline success before a first load is not guaranteed.

Ship an update

- 1 Change a visible version label, update the cache version and test the installed app refresh.
- 2 Check: The new version eventually appears and the old cache is removed.
- 3 Hint: Document the observed update behavior instead of promising instant replacement.

When it goes sideways

01

No install button appears

The browser UX or installability requirements differ. Try: Inspect the manifest and security context, then follow the device-specific install flow.

02

Offline shows a broken page

Assets were never cached or their paths are wrong. Try: Inspect service-worker cache entries and test after a successful online load.

03

The old app persists

An obsolete cache is still serving files. Try: Use an explicit cache version and activation cleanup.

Review your evidence

1

Manifest and both icons validate.

2

App is served in an appropriate security context.

3

A real phone install test is documented.

Review your evidence

1

Offline and online capabilities are distinguished.

2

An update/cache-version test passes.

What does a service worker cache guarantee?

- 1 Every remote API call works offline.
- 2 Only the resources and behavior your caching design actually supports.
- 3 Automatic cross-device sync.

Why test a cache update?

- 1 To prove a new version can replace the old files.
- 2 To increase the icon size.
- 3 To remove the need for HTTPS.

Build with a clear contract

Convert BorrowBox into a small PWA folder. Provide index.html, manifest.webmanifest and sw.js using existing 192/512 PNG icons. Explain localhost development and HTTPS phone testing. Cache the static shell with a named version and remove obsolete caches on activation. Include a manual install, offline and update checklist. Do not claim offline AI or cloud sync.

[Open copyable prompt](#)

Install a field checklist

- 1 Local edits survive reload.
- 2 Network-only behavior is labeled.
- 3 A stale version can be identified.

Build independently, then teach it back

Your next challenge

Adapt the PWA to a volunteer setup checklist and add a visible last-saved indicator.

Explain your decisions

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