GUIDE / RESOURCE

RapidAir Reference Playbook

How the first audit run became a repeatable farming workflow

RoleModel software

RoleModel Software
May 2026

RapidAir is the reference example for turning one strong UX audit into farmable design work. The value of the run was not only in the findings. It proved that a good audit can become a repeatable operating model when the report, design exploration, prototype, and deployment are treated as one connected pipeline.

> **Reference pipeline**
>
> run code → audit → code → Figma → Figma Make → React demo → Vercel

01

Begin with the real product

The audit began by running the product and tracing the primary user journey in context. That first pass kept the work anchored to lived experience instead of isolated screens, and it exposed which moments deserved deeper attention.

01.01

Run the code

A working product gives the auditor the sequence, timing, and friction that static screenshots hide. Future audits should begin with the application running and one documented pass through the core workflow.

01.02

Audit the experience

RapidAir worked best when the audit used consistent lenses but resisted the urge to become a defect dump. The client-facing version became stronger once it told a clear story: what was already good, what expectations had evolved, and what should happen next.

02 Make the findings legible

The first run became useful to others when the findings were turned into a coded report artifact rather than left as loose notes. That artifact created a stable bridge between analysis and design.

02.01

Build the report in code

1. Keep only the sections that the audit can support with a real argument or visible direction.
2. Use confident language. A completed audit should recommend, not hedge.
3. Keep the artifact structurally predictable so later export and reuse stay clean.

The first pass needed manual cleanup around faint text, inconsistent vertical rhythm, and slides that implied design work that did not yet exist. Those corrections became durable rules for the next run.

03

Move from report to design

Figma changed the report from a finished document into a collaborative design object. Figma Make then turned the strongest opportunities into visible interface directions, which kept the audit from stopping at critique.

03.01

Use Figma as the review surface

The import worked best once the coded report used consistent section structure and avoided fragile layout tricks. The export path also taught a practical lesson: confirm the local server root before constructing capture URLs.

03.02

Use Figma Make selectively

> **Selection rule**
>
> Design only the opportunities that strengthen the story. More screens do not automatically create a better audit.

04

Prototype only when interaction matters

The React demo mattered because part of the recommendation lived in motion, sequence, and felt workflow. A static screen could show the destination, but the prototype could prove the route.

- Use a coded demo when interaction itself is the argument.
- Keep the demo separate when it has its own dependencies, history, and release cadence.
- Treat the prototype as a bridge from design intent to implementation conversation.

05

Publish the work without collapsing the system

Vercel made the report and demo shareable, reviewable, and durable after the meeting ended. Publication also forced a healthy repo boundary: reusable workflow code belongs in the skill repo, static report bundles belong in the projects catalog, and larger demos keep their own lifecycle.

> **System boundary**
>
> One pipeline, three homes: reusable skill logic, static report artifacts, and larger interactive demos.

06

The reusable operating model

1. Run the product and document the main journey.
2. Audit through consistent lenses and shape the findings into a clear narrative.
3. Build a coded report artifact that is stable enough to review, export, and reuse.
4. Move into Figma for collaborative review and visible design direction.
5. Prototype only when interaction needs to be demonstrated.
6. Publish the outputs so they remain useful after the meeting.

What future designers should copy

Copy the connected pipeline, the confidence of the narrative, and the discipline around system boundaries. Improve the startup friction, preserve only real sections, and avoid generating design work merely to make the deck feel fuller.