How to Use — React Deep Audit

Version 1.5

A production-readiness audit for AI-built React apps

What This Toolkit Does

You vibe-coded something that works. This toolkit helps you make it **production-ready**.

Most AI-built apps ship with hidden problems:

- exposed API keys
- missing auth / authorization
- fragile architecture
- performance issues that show up under real load
- UX gaps that make users bounce
- App Store rejection landmines

This audit pack finds those problems **before your users do**, and gives you fix prompts you can paste directly into your AI dev assistant.

Who This Is For

This is for you if:

- you built a React / React Native / Expo app using AI assistance
- you're preparing to launch to real users
- you want a clean list of fixes, prioritized by severity
- you want a professional "ship-ready" review process

This is not for you if:

- you're still experimenting early-stage
- you don't plan to ship
- you don't want to fix issues (these audits will find them)

What You Need

You need:

- your project folder on your machine
- access to Claude / Cursor / ChatGPT
- 30–60 minutes to run audits
- time afterward to apply fixes

Recommended:

- **Claude** for strongest analysis
- **Cursor** for fastest fixes

Before You Begin (Important)

1) Commit your code

This gives you a rollback point.

```
git add .
git commit -m "Pre-audit snapshot"
```

2) Copy the audit folder

Place `tg-deep-audit/` in the root of your project.

3) Block off focused time

These audits are high-leverage. Treat them like a launch checklist.

The Workflow (How You Run an Audit)

Each audit is a Markdown prompt file.

For each prompt:

1. Open the prompt file
2. Copy everything
3. Paste into your AI assistant (Claude / Cursor / ChatGPT)
4. Wait 2–5 minutes
5. Review the report and save it
6. Repeat with the next prompt

Every audit generates a results file here:

/audit_results/[AuditName].md

Recommended Audit Order (Run Them Like This)

The audits build on each other. Run them in this sequence:

Critical Foundation

1. Security
2. Code Quality
3. Performance

Data Protection

4. Data Security
5. Third-Party Libraries

User-Facing Quality

6. Mobile Layout
7. User Experience
8. Accessibility

Technical Correctness

9. Date / Time Handling

App Store Readiness

10. App Store Readiness

Understanding Your Results

Each audit produces four things:

1) Report Card

A letter grade + severity breakdown.

2) Issues With Context

Each issue includes:

- file paths
- line numbers
- code snippets
- why it matters
- severity level
- time estimate

3) Ready-to-Use Fix Prompts

These instruct your AI assistant how to fix each issue safely.

4) Prioritized Action Plan

A recommended order so you can fix efficiently.

How to Fix Issues (Best Practice)

Don’t try to fix everything at once.

Instead:

1. Fix 🔴 Critical issues first
2. Then 🟠 High
3. Then 🟡 Medium

After each group of fixes:

- run your tests
- run a production build
- open the app and click through the flows
- commit your changes

A good commit rhythm:

```
git commit -m "Fix: security audit issues"
git commit -m "Fix: performance improvements"
```

How Long This Usually Takes

Typical effort:

- Run all audits: **30 minutes**
- Fix critical issues: **60 minutes**
- Finish the full list: **A couple hours**

Depends on app size and AI code quality.

Common Mistakes (Avoid These)

❌ Running audits out of order
❌ Skipping Security / Data Security
❌ Fixing everything without testing in between
❌ Making huge changes without commits
❌ Ignoring “medium” severity UX issues (users notice them)

If You Only Do Three Audits…

Do these:

1. Security
2. Data Security
3. Third-Party Libraries

Those prevent:

- breaches
- data loss
- surprise bills

Need Help?

If you get stuck on fixes or want hands-on support:

Email: thundergodsoftware@gmail.com

Newsletter: https://thundergodsoftware.substack.com/subscribe

Final Note

AI gets you to 80% fast. The last 20% is where apps either become real… or fall apart.

You’re doing the part most developers skip.

Ship with confidence. 🚀

React Deep Audit
Thunder God Software