

AI App Launch Audit

This sample shows the shape of the PDF report I send after reviewing an AI-built app. The goal is to identify what could damage user trust before launch and put the fixes in a practical order.

PROJECT Customer onboarding flow	SCOPE 3 flows, 5 pages	REVIEWED Prototype + repo	VERDICT Fix before launch
---	---	--	--

Verdict: Fix before launch

Executive Summary

WHAT IS WORKING

- The core onboarding path is understandable and close to shippable.
- The UI handles the happy path clearly on desktop and mobile.
- The product has enough structure to fix the main risks quickly.

WHAT SHOULD CHANGE FIRST

- Fix silent failures in password reset and invite acceptance.
- Prevent duplicate billing confirmation requests.
- Defer dashboard chart code until after the first useful screen.

Top Findings

LAUNCH BLOCKER

PASSWORD RESET FAILS SILENTLY AFTER INVALID INPUT

The form accepts invalid email input and then shows no useful error after submit. A user who mistypes their email will not know whether the product is broken, the account does not exist, or the reset email is delayed.

Recommended fix: validate the input before submit, show a clear error state, and include a successful submitted state that does not reveal whether the account exists.

LAUNCH BLOCKER

BILLING CONFIRMATION CAN CREATE DUPLICATE REQUESTS

Refreshing the confirmation page retries the billing confirmation request. This is risky before launch because it can create duplicate support tickets, inconsistent account state, or duplicate payment provider calls.

Recommended fix: make the confirmation action idempotent, disable repeat submits, and render a stable success page after the first confirmed response.

HIGH PRIORITY

INVITE ACCEPTANCE HAS NO LOADING OR FAILED STATE

The invite acceptance screen looks inactive while the request is in flight. On slower connections, this invites double clicks and makes the first-use experience feel unreliable.

Recommended fix: add a pending state, disable the primary action while submitting, and show a retryable error when the request fails.

Performance Pass

This is not a full lab or field performance audit, but it covers the most obvious launch risks: slow first load, excessive JavaScript, layout shifts, sluggish interactions, and asset handling that could hurt Core Web Vitals.

AREA	OBSERVED RISK	SUGGESTED ACTION
JavaScript	The authenticated dashboard ships the charting bundle before the user sees their first task.	Lazy-load the chart module and render the task list first.
Images	Marketing and onboarding hero images do not reserve stable dimensions, which can cause layout shifts on mobile.	Set explicit image dimensions or an aspect-ratio container.
Interaction	The settings page performs validation and save work on the main thread after every field change.	Debounce validation and move expensive work out of immediate input handlers.

Suggested Fix Order

1. Fix password reset feedback and invite acceptance error states.
2. Make billing confirmation idempotent.
3. Defer the dashboard chart bundle.
4. Reserve image dimensions in onboarding and marketing views.
5. Clean up duplicated auth checks before adding more account features.

Final Verdict

The app is close, but I would not launch the onboarding and billing flows to real users until the two launch blockers are fixed. The remaining items can be handled immediately after launch if the deadline is tight, though the dashboard performance issue is worth fixing before a public demo.

Sample report by Nicholas Ray. This is illustrative only and is not a security audit, compliance review, or guarantee of production readiness.