

Cosmo Feature Walkthrough

Lite + full-build QA guide for sitting down, playing with Cosmo, and hardening the user experience before closed alpha.

Use this like a game test, not a checklist.

Try one path. Screenshot anything confusing, broken, or surprisingly good.

Score each pass: green, rough, broken, or wow.

1. First Launch

Cosmo should feel safe, alive, and easy before it feels technical.

Try	What proves it
-	Launch from a clean install and watch the startup screen for at least 10 seconds.
-	Confirm Cosmo stays inside the startup glass and no black square flash appears.
-	Read only the visible copy. It should explain local-first value without jargon.
-	Capture the first screen and one sentence: I think Cosmo is...

2. Lite Home

The home state should breathe. It should show value without turning into a dashboard maze.

Try	What proves it
-	Collapse and expand the dock, hover the main buttons, then open Cards, Settings, and Community.
-	Check that locked or planned features are honest and easy to understand.
-	Capture the clean home state plus the most confusing button if one exists.

3. Handoff Loop

The core pitch is simple: Cosmo notices, stages, hands off, and waits.

Try	What proves it
-	Open the CNS / handoff folder button.
-	Find the staged prompt or packet flow.
-	Confirm nothing runs automatically and the user always approves.
-	Capture the view that best explains Claude, Codex, local model, and user roles.



4. Learning Glow

The user should see local learning without fake cloud-training claims.

Try	What proves it
-	Open Learning Glow and read the labels.
-	Find proof or explanation for what each signal means.
-	Check that empty learning data gives a useful next step instead of dead UI.

5. Workflow Studio Preview

This should feel like a safe sketchpad for automation ideas.

Try	What proves it
-	Open Workflow Studio, inspect the first template, and stage a card if available.
-	Verify it says stage-only and does not run commands.
-	Capture the first card you naturally want to click.

6. Widgets

Widgets should be useful, movable, and fun without cluttering the desktop.

Try	What proves it
-	Try notes/document companion, media/community widget, or any available surface widget.
-	Move, resize, collapse, stamp, or return it if those controls exist.
-	Confirm widgets do not trap Cosmo or hide important controls.

7. Settings

Settings should feel like customization cards, not a wall of switches.

Try	What proves it
-	Find motion/effects/audio, privacy, and feature availability.
-	Check that unavailable features are greyed out with a useful reason.
-	Capture the clearest and most confusing settings groups.

8. Discord And Community

Community should feel optional, useful, and alive.

Try	What proves it
-----	----------------

-	Find the permanent invite, feedback path, roadmap, and free-vs-paid explanation.
-	Verify no login or token is required for Lite community links.
-	Capture one sentence: The community gives me...

9. Full Build Preview

Lite should create desire without pretending locked features are active.

Try	What proves it
-	Inspect full-build previews like LOOM, local model, project browser, terminal, or advanced workflows.
-	Check active, preview, locked, and planned labels.
-	Capture one feature you would upgrade for.

10. Close And Reopen

Cosmo should remember enough without leaving a mess.

Try	What proves it
-	Close Cosmo, reopen it, and confirm it returns to a clean state.
-	Check for zombie terminal/window behavior, repeated first-run prompts, or broken widget bounds.
-	Do not hand to testers if this pass fails.



Score Sheet

Use one line per pass. The goal is not to be polite. The goal is to find friction before users do.

Area	Green	Rough	Broken	Wow	Note
First launch					
Lite home					
Handoff loop					
Learning Glow					
Workflow Studio					
Widgets					
Settings					
Discord					
Full preview					
Reopen					

Stop Conditions

Do not hand Cosmo Lite to a tester if first launch crashes, Cosmo escapes the startup orb, uninstall is unclear, Lite exposes advanced terminal/model features by accident, public copy overclaims proof, or OAuth/provider/social posting appears enabled by default.