That’s a great question — and it gets at the heart of what Streamlit Cloud is versus what it isn’t.

**✅ What Streamlit Cloud is good for**

* Rapid prototyping / MVPs: Perfect for quickly showing an interactive demo (like CleanList) without worrying about infrastructure.
* Data apps & dashboards: Many analysts and data scientists host dashboards here that only need light traffic.
* Small teams / pilot testing: Easy to share with colleagues, advisors, or a small pool of testers.

Think of it like Airbnb for prototypes → quick, cheap, frictionless hosting to validate an idea.

**❌ Where Streamlit Cloud falls short**

* Scalability: It’s not designed for thousands of concurrent users. Performance bottlenecks show up quickly.
* Custom domains & branding: Limited customization compared to a proper deployment on AWS, GCP, or Azure.
* Advanced integrations: Stripe, Firebase, background jobs, and heavy APIs often require more control than Streamlit Cloud gives.

So if you’re aiming for random public testing at scale (hundreds/thousands of users), Streamlit Cloud can’t sustain that long-term.

**🚀 What successful apps usually do**

Most apps that start on Streamlit Cloud eventually:

1. Prove their concept (get feedback, validate demand).
2. Migrate to a more scalable stack: e.g.
   * Deploy Streamlit on Heroku, Render, Railway (easy to scale, cheap starter plans).
   * Or containerize with Docker + AWS ECS/Fargate for full production control.
   * Or rewrite the frontend in React/Next.js while keeping Streamlit as an internal tool.

**🎯 For CleanList**

* Streamlit Cloud = Phase 1: Great for demos, small-scale testing with friends/investors.
* Phase 2 = Scalable hosting: Once you want “random testers” or to onboard early adopters at scale, you’ll need to migrate.

👉 In other words: Streamlit Cloud is a runway, not the airplane hangar. It gets you off the ground quickly, but you don’t fly coast-to-coast on it.

Would you like me to map out a graduation path for CleanList — i.e., when to know it’s time to move from Streamlit Cloud → Heroku/Render → full-scale production?

React/Next.js (customer-facing frontend)

* + Polished, mobile-responsive, branded website + app experience.
  + Easy to integrate advanced features: push notifications, PWA, SEO, animations.
  + Lets you look & feel like a consumer app, not a data demo.

✅ The winning playbook I’ve seen:

* Use Streamlit Cloud → MVP demo.
* Migrate to Render/Heroku → broader testing.
* Then, split: React/Next.js frontend for users, keep Streamlit for ops.