## **2. Deployment Options**

### **A. Streamlit Community Cloud (free, quick)**

Push your code to a public GitHub repo.

Sign in at share.streamlit.io with GitHub.

Deploy directly from the repo — your app will be public.

Pros: Fast, free, no infra setup.

Cons: Limited resources; code is visible if repo is public.

### **B. Hugging Face Spaces (Gradio or Streamlit)**

Similar to Streamlit Cloud — you can host Streamlit apps for free.

Supports both **public** and **private** spaces.

Pros: Integrates well with ML demos; easy to share.

Cons: Free tier has limited compute; private spaces require paid tier.

### **C. Internal Server / VM**

Run streamlit run app.py on an internal cloud machine or VM.

Use your company network or VPN for access.

Pros: Keeps data/code internal; more control.

Cons: Requires infra setup and network access.

### **D. Docker + Cloud Hosting**

Wrap the app in a Docker container and deploy to AWS, Azure, GCP, or Heroku.

Pros: Fully customizable, can scale, private or public.

Cons: More DevOps work than pure Streamlit hosting.

## **3. MVP Deployment Checklist**

Before going live, especially if other people will use it:

**Dependencies** — Add a requirements.txt listing all Python packages (pip freeze > requirements.txt).

**Data Privacy** — If people upload lists, decide if you store them or process only in memory.

**Usage Limits** — For public demos, set sensible file size limits.

**Tier Threshold Defaults** — Ensure sensible defaults for CleanScore tiers.

**Branding/Disclaimer** — Add a “Not medical advice” note, which the sample already has.

## **4. Recommendation**

If your goal is to share quickly with a few testers or stakeholders:

Use **Streamlit Cloud** or **Hugging Face Spaces** for the MVP.

Keep everything in one GitHub repo with app.py and requirements.txt.

Add basic instructions to the repo README.

**Practical Playbook**

* ✅ Keep building product logic in Streamlit.
* ✅ Add custom domain + responsive design to make it customer-facing.
* ✅ Layer in auth, payments, PWA support for a minimal viable product.
* ✅ Later, if you need more polish or scalability, you can rebuild UI in React/Next.js but keep the backend logic.

If your goal is **internal-only** (especially with any sensitive grocery data), run it inside your company’s cloud network or containerize it for secure hosting.