**ready-to-deploy repo** you can copy/paste. It assumes you’ll use the **Streamlit app code I gave you earlier** (save that as app.py). Everything else below is new and optimized for fast deployment.

# **📁 Repository layout**

grocery-cleanscore-app/  
├─ app.py # <- paste the Streamlit app code you already have  
├─ requirements.txt  
├─ README.md  
├─ .gitignore  
├─ .streamlit/  
│ └─ config.toml  
├─ Dockerfile # optional (for AWS/GCP/Azure/VMs)  
└─ Procfile # optional (for Heroku-style platforms)

## **📦 requirements.txt**

Pinning reasonable, widely compatible versions:

streamlit>=1.34,<2.0  
pandas>=2.0,<3.0  
openpyxl>=3.1,<4.0

If you add anything else later (e.g., numpy), re-export with pip freeze > requirements.txt before deploying.

## **🧹 .gitignore**

# Python  
\_\_pycache\_\_/  
\*.pyc  
.venv/  
venv/  
.env  
  
# Streamlit  
.streamlit/secrets.toml  
  
# OS / editors  
.DS\_Store  
Thumbs.db  
.idea/  
.vscode/

## **🎛️ .streamlit/config.toml**

This keeps the UI clean and predictable in the cloud:

[server]  
headless = true  
enableCORS = true  
enableXsrfProtection = true  
port = 8501  
  
[browser]  
gatherUsageStats = false  
  
[theme]  
base="light"  
primaryColor="#2F855A"  
backgroundColor="#FFFFFF"  
secondaryBackgroundColor="#F7FAFC"  
textColor="#111827"

## **🐳 (Optional) Dockerfile**

Use this if you’re deploying to a VM, ECS, Cloud Run, etc.

FROM python:3.11-slim  
  
# System deps (faster openpyxl wheels; add build tools only if needed)  
RUN apt-get update && apt-get install -y --no-install-recommends \  
 build-essential \  
 && rm -rf /var/lib/apt/lists/\*  
  
WORKDIR /app  
  
# Copy dependency spec first (better layer caching)  
COPY requirements.txt ./  
RUN pip install --no-cache-dir -r requirements.txt  
  
# Copy app  
COPY . .  
  
# Streamlit needs this env for containerized runs  
ENV PYTHONUNBUFFERED=1 \  
 PIP\_NO\_CACHE\_DIR=1  
  
EXPOSE 8501  
CMD ["streamlit", "run", "app.py", "--server.port=8501", "--server.address=0.0.0.0"]

## **🔧 (Optional) Procfile**

For Heroku-style platforms:

web: streamlit run app.py --server.port=$PORT --server.address=0.0.0.0

# **🚀 Deploy options**

## **Option A — Streamlit Community Cloud (fastest public MVP)**

1. Push the repo to GitHub (public).
2. Go to <https://share.streamlit.io/> and connect your GitHub.
3. Select the repo, set the entrypoint to app.py.
4. The app builds automatically using requirements.txt.

**Tips**

* Keep uploads small (Streamlit Cloud is resource-limited).
* If you ever see a memory error, tighten file size/row limits or move to Docker on a beefier host.

## **Option B — Hugging Face Spaces (Streamlit app)**

1. Create a new Space → **SDK: Streamlit** → choose public or private.
2. Upload all files (app.py, requirements.txt, etc.).
3. The Space auto-builds and hosts your app.

**Tip**: Private Spaces require a paid plan.

## **Option C — Internal VM / Company Cloud**

**With Docker**

docker build -t grocery-cleanscore:latest .  
docker run -p 8501:8501 grocery-cleanscore:latest

Open [http://localhost:8501](http://localhost:8501/) (or map through your load balancer/VPN).

**Without Docker**

python -m venv .venv  
. .venv/Scripts/activate # Windows  
# or: source .venv/bin/activate # macOS/Linux  
pip install -r requirements.txt  
streamlit run app.py

# **🔐 Privacy & MVP hardening**

* **Uploads**: The current app processes files in memory and doesn’t persist them — good for MVP. If you later add storage, document retention clearly.
* **Limits**: Add basic size guards (e.g., reject files > 5–10 MB) if you expect the public to test it.
* **Disclaimers**: Keep the “not medical advice” footer (already included).
* **Monitoring**: For internal deploys, consider reverse proxy + basic auth.

# **✅ Quick local run (dev)**

# from the repo root  
python -m venv .venv  
# Windows:  
. .venv/Scripts/activate  
# macOS/Linux:  
# source .venv/bin/activate  
  
pip install -r requirements.txt  
streamlit run app.py