Product Requirements Document (PRD)

Project: Dose & Vial Optimizer (HCP Calculator — Streamlit + GitHub Deployment)
Owner: Mithun Aggarwal
Version: 1.0
Date: 4th Oct 2025

1. Background / Context

Healthcare Providers (HCPs) prescribing a new oncology treatment need to quickly calculate patient dose requirements and determine the optimal combination of two available vial strengths (70 mg and 100 mg).

The business goal is to provide a **simple, accurate, web-based calculator** for HCPs that:

- Accepts patient weight and regimen type (2.5 mg/kg starting, 1.9 mg/kg reduced).
- Calculates the required dose (rounded up to the nearest mg).
- Shows the minimal-waste vial combinations using only 70 mg and 100 mg vials.
- Clearly highlights the **best option** (least waste → fewest vials as tie-breaker).
- Provides export/print functionality.
- Uses a **GSK-inspired theme** (orange + white) for consistent branding.
- Runs via **Streamlit**, deployable on **GitHub + Streamlit Community Cloud**.

2. Objectives

- Deliver a **fully functioning Streamlit application** embedding the existing HTML/JS calculator logic.
- Wrap it in a professional repository with:
  - Proper Python/Streamlit app entrypoint (`streamlit_app.py`).
  - Static HTML file (`app_static.html`) containing the updated calculator UI & logic.
  - Dependency management (`requirements.txt`).
  - Configurable branding (`.streamlit/config.toml`).
  - Documentation (`README.md`).
- Allow seamless deployment to Streamlit Community Cloud from GitHub.
- Ensure maintainability so non-technical teams can edit the static HTML later.

3. Scope

In-Scope

- Conversion of existing static HTML/JS into a **Streamlit-wrapped app**.
- GitHub repository setup with standard best practices.
- Orange/white styling updates per GSK branding guidelines.
- CI/CD via Streamlit Cloud auto-deploy from GitHub.
- Local testing instructions.
- Self-test cases embedded in JS remain accessible to developers.

Out-of-Scope

- Any backend database or API integration.
- Multi-tenant login, authentication, or user analytics.
- Regulatory submission—tool is strictly **internal HCP aid**.

---

4. Functional Requirements

1. **Inputs**
   - Weight (kg, numeric, min=0, step=0.1).
   - Regimen (radio buttons: 2.5 mg/kg or 1.9 mg/kg).
   - Vial strengths are fixed (70 mg, 100 mg) — shown but **not selectable**.
   - Max total vials (default = 10, adjustable).
2. **Processing**
   - Required dose = `ceil(weight * regimen)` (rounded up).
   - Enumerate vial combinations:
     - Only 70 mg and 100 mg vials allowed.
     - No underdosing.
     - Waste = (total – required).
     - Waste % = waste / total.
   - Ranking:
     - Least waste first.
     - Tie-breaker: fewer vials.
3. **Outputs**
   - Summary: "X mg/kg × Y kg → required Z mg (rounded up). Two vial strengths: 70 mg & 100 mg."
   - Required dose shown in info banner.
   - Recommendation: "Exact match" or "Min waste option → order A×70 mg + B×100 mg …"
   - Table of up to **4 options**:
     - Columns: 70 mg vials, 100 mg vials, total mg, waste mg, waste %, total vials.
     - Highlight best option with badges ("Least waste", "Fewest vials").
4. **Actions**
   - Calculate options (primary CTA).
   - Reset (clear inputs, back to default).
   - Print / Save as PDF.
5. **Styling**

    - GSK Orange & White theme:
        - Background: White (#ffffff).
        - Accent/CTA: Orange (#ff6a00).
        - Cards: light orange (#fff5ef).
        - Muted labels: burnt orange (#8a4a10).
    - Table rows in white with soft grey dividers.
    - Clear info banners for dose summary and recommendation.

---

5. Non-Functional Requirements

- **Performance:** Results calculate instantly (<100ms client-side).
- **Usability:** Mobile-responsive within Streamlit iframe.
- **Maintainability:** Core HTML calculator remains independent (`app_static.html`).
- **Compliance:** Banner disclaimer must always display:
  “For HCP guidance only. Verify calculations against the product label and local policy.”

---

6. Technical Requirements

- Language: Python 3.10+
- Framework: Streamlit (latest stable)
- Dependencies: `streamlit>=1.33`
- Repo host: GitHub
- Deployment: Streamlit Community Cloud

Repo structure:

```
dose-vial-optimizer/
├─ streamlit_app.py
├─ app_static.html
├─ requirements.txt
├─ README.md
└─ .streamlit/
   └─ config.toml
```

`.streamlit/config.toml` example:

```
[theme]
base=\"light\"
primaryColor=\"#ff6a00\"
backgroundColor=\"#ffffff\"
secondaryBackgroundColor=\"#fff5ef\"
textColor=\"#2b2b2b\"
```

---

7. Success Criteria

- App runs locally via `streamlit run streamlit_app.py`.
- Deploys successfully to Streamlit Community Cloud directly from GitHub.
- Users can enter weight/regimen, calculate, and receive clear results in <1s.
- Visual branding matches GSK orange/white theme.
- No ability to modify vial strengths.

---

8. Future Enhancements (Phase 2)

- Multi-language support.
- Export to CSV in addition to PDF.
- HCP login to save patient scenarios.
- Integration with e-detailing systems or Salesforce.

✅ This PRD gives Codex **clear technical specs + repo design + success criteria.**

Do you want me to also **generate the starter repo files (streamlit_app.py, requirements.txt, config.toml, README.md)** as part of the PRD so Codex can just code-complete them?