Product Requirements Document (PRD)

Project Name: Nonprofit Data Collector (ProPublica → Google Sheets)

Prepared For: Upwork Developer Engagement

Version: v1.0

---

1. Overview

We need a reusable tool that continuously collects nonprofit organization data from the **ProPublica Nonprofit Explorer API** and saves it into a **Google Sheet**.

The system should:

- Always pull from the **latest Form 990 filing**.
- Apply filtering rules so only relevant organizations are kept.
- Append only **new results** to the dataset, avoiding duplicates.
- Allow me to **configure keywords and contribution ranges** to control the results.
- Be reliable and easy to re-run as often as needed.

The end goal: a growing, deduplicated dataset of nonprofits that can be expanded in batches (200–500+ orgs per run).

---

2. Objectives

- **Accuracy:** Ensure only nonprofits matching defined filters are included.
- **Reusability:** Script/app can be run repeatedly without duplicating EINs.
- **Flexibility:** Keyword list and filters are easily editable.

- **Simplicity:** Results appear directly in a Google Sheet — no manual file uploads.
- **Scalability:** Handle 200–500 new orgs per run, with option to scale higher.

---

3. Scope

In-Scope

- Fetch nonprofits via ProPublica API keyword search.
- Fetch org details to determine eligibility.
- Always use **latest Form 990 or 990-EZ filing** (exclude 990-PF).
- Apply contribution, type, and deduplication filters.
- Append results to Google Sheets (one master sheet).
- Provide a **Config sheet or config file** for keywords, ranges, toggles.
- Handle errors gracefully with retries/backoff.

Out-of-Scope (future iterations)

- Advanced UI/UX (beyond basic config via Google Sheets or CLI).
- Political/issue tagging (optional for later).
- Automatic enrichment (emails, social, etc.).

---

4. Functional Requirements

4.1 Data Source

- **API:** ProPublica Nonprofits API v2.

- Endpoints:
    - /search.json?q={keyword}&page={n}
    - /organizations/{EIN}.json

4.2 Filters

- Contributions & Grants: **$20M – $500M**.
- Exclude private foundations (formtype == "990PF").
- Exclude schools/education:
    - NTEE codes starting with “B”.
    - Names containing: university|college|academy|school|seminar(y|ies).
- Exclude religious organizations:
    - NTEE codes starting with “X”.
    - Names containing: church|diocese|synagogue|mosque|minist(ry|ries)|bible|christ|catholic|presbyterian|lutheran|baptist.
- Deduplicate EINs (both within a single run and across all runs).

4.3 Data Fields (columns in Google Sheet)

- name
- ein
- fy_end (fiscal year end)
- formtype (e.g., 990, 990EZ)
- ntee_code
- contributions_and_grants (integer)

- website (from org profile or parsed from XML)
- filing_url (PDF if available, else XML)
- source (always “ProPublica”)
- pulled_at (ISO timestamp)
- amended (true/false if filing is amended)

4.4 Configurability

- Keywords (list editable in Config sheet).
- Min/max contribution range.
- Max results per run (e.g., 200, 500, 1000).
- Pages per keyword.
- Toggle: exclude hospitals.

4.5 Google Sheets Behavior

- Append new rows only if EIN not already present.
- If EIN already exists, **update** row only if newer filing is found.
- Maintain a second **Config tab** for inputs (keywords, ranges, toggles).
- Optionally, maintain a **Logs tab** with summary of each run.

---

5. Non-Functional Requirements

- **Reliability:** Retries/backoff on 429/500 errors.
- **Politeness:** Delay ~0.1–0.2s between API requests.

- **Performance:** Should complete 200–500 orgs in a reasonable runtime (<15 minutes).
- **Maintainability:** Clear documentation (README) for setup and use.
- **Transparency:** Logging that explains why an org was skipped.

---

6. Example Workflow

1. Developer runs script (locally or via scheduled job).
2. Script reads keyword list from Config sheet.
3. For each keyword:
    - Query ProPublica search endpoint (multiple pages).
    - Fetch each org's detail.
    - Select latest Form 990/990EZ.
    - Apply filters.
    - Capture required fields.
4. Append deduped results to Google Sheet.
5. Print console summary:
    - Keywords searched.
    - Orgs added.
    - Orgs skipped (reason).
    - Total in sheet.

---

7. Deliverables

- **Script/application** (Python preferred).
- **Google Sheets integration** (with OAuth flow and token reuse).
- **Documentation** (setup + usage).
- **Demo run**: At least 50 orgs written into a provided Google Sheet.

---

8. Nice-to-Haves (Future)

- Issue/cause tagging (e.g., veterans, Second Amendment, climate).
- Resume from last run (store cursors).
- Lightweight web UI (e.g., Streamlit) for non-technical users.

---

9. Acceptance Criteria

- I can run the tool multiple times, and the Google Sheet continues to grow with no duplicate EINs.
- Each org's data reflects its **latest available 990 filing**.
- Filtering works (all results within range, no private foundations, no schools/churches).
- Keywords are editable by me.
- Output is clean, structured, and usable.