Product Requirements Document (PRD)

Project Title: Political Campaign Finance Tracker

Author: Trey Davis

Date: September 17, 2025

Status: Draft / Proposed

1.0 Executive Summary

This project aims to build a publicly accessible, data-driven platform that provides a clear and transparent view of campaign finance data for U.S. federal elections. The goal is to make a complex and often opaque dataset understandable to the average person, empowering journalists, researchers, and citizens with the information they need to hold politicians and large donors accountable. The core of this product will be a robust, automated data pipeline and a well-structured database that can be used to power a variety of analytical dashboards and reports.

2.0 Vision and Goals

- **Vision:** To democratize access to campaign finance data, transforming raw, messy government data into an accessible and powerful tool for civic engagement.
- **Business Goals (for your portfolio):**
    - Demonstrate a strong understanding of the entire data engineering lifecycle, from data acquisition and cleaning to modeling and visualization.
    - Showcase proficiency with key technologies (e.g., Python, SQL, a specific database).
    - Create a compelling and unique portfolio project that stands out to potential employers.
    - Produce a tangible, public-facing artifact (dashboard or report) that highlights key insights.

3.0 Target Audience / User Personas

1. **The Curious Citizen:**
    - **Need:** Wants to know who is funding their local or national political candidates.
    - **Questions:** "Who gave money to my representative?" "How much money did they raise from my state?"
    - **User Story:** "As a curious citizen, I want to easily search for a politician and see a list of their top donors so that I can understand their funding sources."
2. **The Student / Researcher:**
    - **Need:** Needs clean, structured data for academic projects or research papers.

    - **Questions:** "What is the average contribution size for a specific party?" "How do contributions from the tech industry compare to the energy sector?"
    - **User Story:** "As a researcher, I need to access a well-documented and consistent dataset so that I can perform my own analysis without spending hours cleaning the data myself."
3. **The Data Journalist:**
    - **Need:** Wants to quickly find compelling stories and trends in the data to create articles.
    - **Questions:** "Which new PACs are receiving the most corporate funding?" "How has the funding landscape for a specific policy issue changed over the last decade?"
    - **User Story:** "As a journalist, I want a dashboard that visualizes key trends and allows me to filter by industry and contribution type, helping me find a lead for a new story."

4.0 Functional Requirements

- **Data Ingestion:**
    - The system must be able to regularly ingest campaign finance data from the Federal Election Commission (FEC).
    - The pipeline should be able to handle both individual contributor data and committee financial data.
    - The pipeline must be idempotent (running it multiple times with the same input data should not create duplicate records).
- **Data Transformation:**
    - The system must clean and normalize data from the raw FEC source. This includes standardizing names, dates, and handling missing values.
    - The system must enrich the contribution data with additional information (e.g., categorizing contributors by industry).
- **Database Schema:**
    - The database schema must be relational and well-documented.
    - It must model the relationships between Candidates, Committees, Contributors, and Contributions accurately.
    - It should be optimized for analytical queries (fast reads) rather than frequent writes.
- **Data Analysis and Querying:**
    - The database must be able to support complex SQL queries to answer the questions outlined in the user stories.
    - It must support aggregation and grouping of data by various dimensions (e.g., date, industry, state, contribution type).
- **Public-Facing Component:**
    - A dashboard must be created to visualize key insights.
    - The dashboard should allow users to filter data by candidate, election cycle, state, and industry.

    - It should present key metrics clearly, such as total funds raised, top contributors, and a breakdown of contribution types (e.g., individual vs. PAC).

5.0 Technical Requirements

- **Platform:** Cloud-based (e.g., AWS, GCP, or Azure) to demonstrate modern cloud skills.
- **Data Source:** Federal Election Commission (FEC) API and bulk data files.
- **ETL/ELT Tooling:**
    - **Programming Language:** Python is the primary choice for scripting the data pipeline.
    - **Libraries:** requests for API interaction, pandas for data manipulation, and a database connector library (e.g., psycopg2 for PostgreSQL).
    - **Orchestration (Stretch Goal):** Apache Airflow or Prefect to automate and schedule the pipeline runs.
- **Database:** PostgreSQL is the recommended choice for a relational database. It is powerful, open-source, and widely used.
- **Visualization/BI Tool:** Tableau Public, Power BI, or a Python-based library like Plotly Dash or Streamlit for creating the dashboard.
- **Version Control:** The entire project code must be managed in a public GitHub repository. The README file should clearly explain the project and its architecture.

6.0 Success Metrics

- **Completion:** The project is a success if it can successfully ingest a full election cycle's worth of data and present a functional dashboard.
- **Documentation:** Success is defined by the clarity of the project's documentation, including the database schema and the ETL pipeline code on GitHub.
- **Insights:** Success is measured by the ability of the project to produce novel and interesting insights that go beyond what a simple search on the FEC website would reveal.
- **Feedback:** The project's success can be measured by the positive feedback received from a community of peers or potential employers.

7.0 Out of Scope

- This project will **not** include a real-time data streaming component. Batch processing of data on a daily or weekly schedule is sufficient.
- This project will **not** include a full-fledged web application with user authentication. The public-facing component will be a simple, static dashboard.
- This project will **not** cover all historical FEC data from all time. The scope will be limited to the most recent 2-3 federal election cycles to keep the project manageable.
- The project will **not** include a "how to" guide for a non-technical audience. The focus is on the technical implementation.