Open Disclosure - Project Overview

Tl;dr: This document is a FPOC for all questions and content related to Open Disclosure.
Document Owners: Ryan Walek (rwalek668@gmail.com), Add your name here...

MEETING NOTES

What is Open Disclosure?

Available at: https://open-disclosure.codeforsanjose.org/
Github available at: https://github.com/codeforsanjose/open-disclosure
City of San Jose contact: Adrian G (Contact rwalek668@gmail.com to get Adrian's contact info)

Open Disclosure is a web app to track campaign finances for the California Primary Election (March 3, 2020), and the General Election (November 3, 2020). The goal of Open Disclosure is to help voters understand who/what Political Action Committees (PACs) are donating money to the candidates/measures.

Some insights that the website aims to expose include:

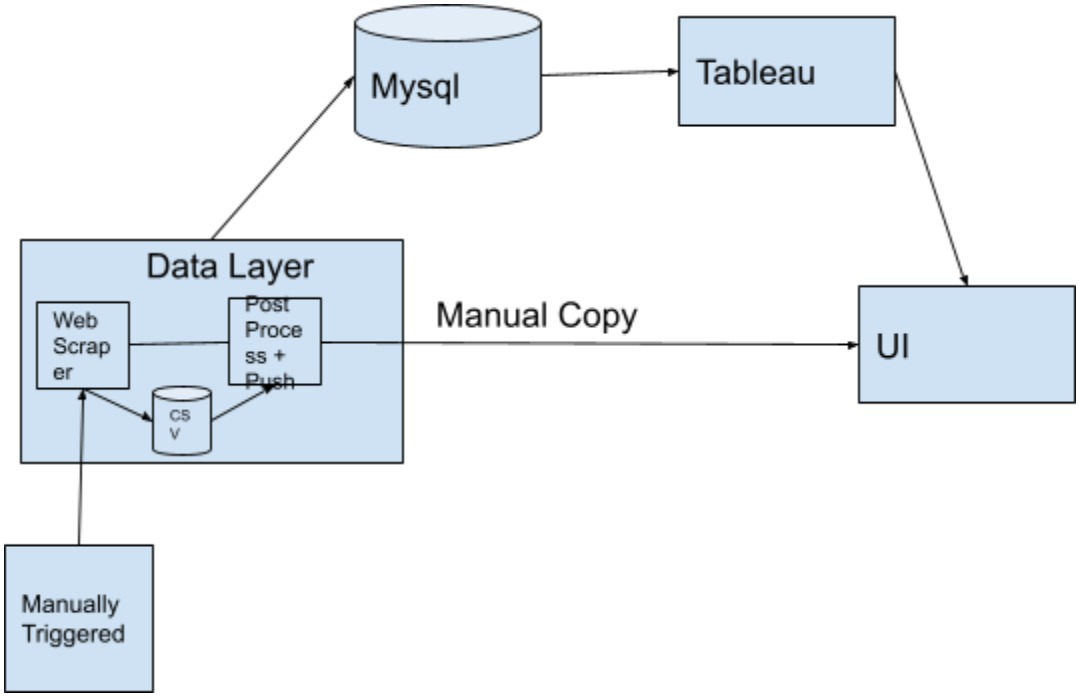
- Are the donors from the same jurisdiction (city/county/state) as the candidate's intended office or outside of the jurisdiction?
- Are the donors individuals, or Political Action Committees (PACs)?
- What is the donor history of the PAC(s)?

We are inspired by Open Oakland's Open Disclosure: https://www.opendisclosure.io
More information can be found about California elections here.

HOW TO UPDATE FOR NEW ELECTION CYCLE
Use this change as an example for the changes you need in your election cycle. After changing the backend, you will have to run the scraper, and manually poke around in the data to verify there is nothing majorly wrong with the conclusions the logic is drawing. You may also have to change the way election data is pulled by the fronted into this file. To do that, simply run aggregatedcsv2redis, grab the output, and manually copy it into this file.

HOW TO DEPLOY
See Github

Components of the Application



1) Note that MySql & Tableau are both outdated.

Data Pipeline

The Scraper

The scraper code, found here. It will output a file in the aggregated_data directory which is the finalized output. This spreadsheet demonstrates the finalized shape of the CSV file.

Note: the full list of files on the website will contain duplicates due to the way addendums are filed. These are filtered out when we choose which files to download in the scraper itself.

Note: The scraper does not currently work in headless mode, meaning it cannot currently be run on the redis server.

Running instructions on github

The Post-Processor

Once the scraper has downloaded the latest raw data and aggregated it, the post-processor is responsible for shaping the data and then uploading it into Redis. The code can be found here. Before running the post-processor, you will need to decide which redis instance to save the data into. Please see the redis section below for details on how to set up redis. By default, the processor will point to a local redis instance.

Running instructions on github

Frontend/UI

- **Running instructions on github**

The Data

Sample Data:
https://docs.google.com/spreadsheets/d/1OuFAqp2ir5l-p19ku8hsK7fHRN61P-UnofeSCBh1aPI/edit#gid=1846868882