Web of Influence Summer Research Project Report - Savio Concessio

Tasks Completed:

- Understanding Requirements

To provide the product, the requirements of the stakeholders were required. Through several meeting we were able to extract these requirements. These requirements helped guide our development, both in including these as features of the product but also as a reference to leave the project ready for requirements the product was unable to be ready for currently. These requirements also developed throughout the project so constantly receiving feedback and consulting the stakeholders helped guide our work throughout.

- Learning Skills

To begin the project, I did not have the necessary specific technical skills to begin development. I spent time developing skills relevant to the project. This included database development, API development, simple front-end using React.js, Figma for UX designs, D3 as a visualization option. This initial learning set the stage for further learning through practice as I developed the product.

- **Planning & Prototyping**

To initialise the development of the product, I did planning on the design of the initial database and worked with Emily prototyping on Figma potential outputs for the website. This allowed us to present our ideas and get feedback on the initial vision for the product. From there the plans were expanded. Along with this I wrote initial prototypes of the database, with isolated tables for the *candidate overview* data. This was chosen as it was the only data with some sort of relative completeness and structure for parsing. This include writing a simple parser in Python to read the csv files and a loader to put it in a local MySQL database. This was tested with local *curl* calls and manual verification with the tables. I then prototyped a basic front-end using React. This contained a few search options and year selections and once searched the API outputs were displayed on the page.

- Extending the Prototype for Mid-Year break

To finish the year, I wanted to have a (basic) working product for use. To do this I

- Restructured and reimplemented the overview data with an external reference to *Party* and *People* tables. This was done to allow for the *View candidate/person* from multiple perspectives/context requirement and to allow for candidates moving parties over the years.
- Updated the API and developed it for filtering. This is so the API can allow for multiple types of arguments, so it is as flexible as possible for future use.
- Cleaned up the UI for a cleaner filter menu, table displayed in a structured way (Emily later made it so it could be viewed in multiple tabs), and general clean ups/improvements.
- Had a little extra time so added *Export outputs to CSV* option in a working capacity.

- Developing the Product for Finish

Once I returned, I started to develop the product more. The first task was building a visualisation. Emily and I made a dynamic bar chart to display the candidate overview data output from the filter. This also had tags and colouring with extra information along with a dynamic legend to select/inspect specific outputs. For the rest of the project, I focused on the wider scope of the project. This included expanding building a loader for the ministerial diaries data and the donations logged. This included linking the new databases to the previous information which ended up requiring a full structure evaluation and adaption due to the limitation of the data that was in these tables. Then the front-end was expanded to account for the expanded data, including a welcome page, candidate overview page, election overview filtering page, ministerial diaries filtering page and a candidate donations log section.

Finished Product:

The project successfully is able to load in the relevant CSV files into the database locally and is flexible for new files to be loaded in to expand the current information. The API is set up for the localhost and works once the data is loaded. The website runs on localhost and allows for the user to search the current information in the database along with the filtered searches.

Further Direction:

Thing to improve

- Improved data quality: The database is missing some information due to the structure of the data we were provided.

- Standing it up: The database, API and website can be hosted on an external service for web access to the product. This would also require tweaks to the API and the front-end code for the new access links and to logging in for the database, but the actual structure of the program is ready for it.
- Improved UI: Currently the UI for some pages is not complete. This was a lower priority task given the timescale of the product as quality product and output was put first, so the output of the website is not complete and should be improved for usability and visual appeal.

Dependencies:

For Mac, running these commands will get the full product on your computer running locally

1. MySQL Community - https://dev.mysql.com/downloads/mysql/
2. Complete the download and set the password
3. Use these commands to access MySQL easily (optional)
    a. echo 'export PATH="/usr/local/mysql/bin:$PATH"' >> ~/.zshrc
    b. source ~/.zshrc
4. Update password on **ALL** python loaders in the file *TestDbLoader* to set MySQL password set earlier (CTR+F 'passwd=' to find all occurrences)
5. https://brew.sh/ for mac
6. Run pip3 install mysql.connector pandas tabulate mysql-connector-python requests flask flask_cors
7. Run *run.py*
8. Run *database_api.py*
9. On the Command Line
    a. Cd demo-cand
    b. npm start
    c. npm install react-router-dom react-chartjs-2 chart.js react-responsive-pagination
    d. npm run dev
10. The website is now running on the localhost link provided