Requirements
Understanding the Web of Influence of People in the NZ Political System

Functional Requirements

Must Have

- The system must allow users to search and filter political donations by name, year, party, and electorate.
- The platform must connect and visualise relationships across datasets (donations, ministerial diaries, appointments, parliamentary records).
- The system must integrate with Snowflake to store, and query structured political data.
- The prototype must support basic user interaction tasks such as tracing donations or identifying connections.
- The platform must log and store user interaction data (anonymised) for evaluation purposes.

Should Have

- The platform should include a graph-based visualisation of relationships between people and political entities.
- The interface should allow users to view detailed data (e.g., individual donations, meeting history) on-demand.
- The system should allow upload of new datasets by an admin or client for ingestion and linking.

Could Have

- The tool could allow users to save and share visualisation views.
- The platform could enable advanced filtering such as by region, board memberships, or keyword tags.
- The interface could include data comparison tools for entities across time.

Won't Have (for now)

- The system won't support real-time data ingestion from government APIs in the initial prototype.
- It won't include automated fact-checking or misinformation detection features.

Non-Functional Requirements

Must Have

- The system must be usable on common browsers (Chrome, Firefox, Edge).
- The application must be secure with password-protected access for editing or data upload functions.
- The system must ensure data privacy, with sensitive logs and user information anonymised.

- The frontend and backend must be responsive and scalable via cloud deployment.

Should Have

- The platform should load results in under 3 seconds for typical user queries.
- The system should support concurrent use by at least 10 users.
- The user interface should be accessible.

Could Have

- The application could support mobile or tablet interfaces.
- The system could include multilingual support.

Won’t Have (for now)

- The system won’t be available as a downloadable desktop application.
- It won’t provide a public API during the prototype phase.