# **Planning Inspectorate**

Consultee List Project

We approached consultation list project by implementing a web app with a seamless user interface to a database. Our tech stack is a java spring app using a postgres database. The front end of the application is built using HTML, CSS & JavaScript, we also heavily used Material Design Lite library to style the web app.

Here is a list of features implemented in this project:

* Deployed a Postgres database
  + Normalised the complicated excel sheet into an appropriate schema that can be deployed as a database. (The schema is shown in the documentation).
* Implemented edit add and delete record functionality
  + This database is accessible via a website utilising an API backend written in Spring Boot.
  + This allows for all CRUD (Create, Read, Update, Delete)
* Implemented search and search filter feature to find specific records
  + Using a variety of database functions, the database can be searched via column and queried from the web interface allowing for easy access to specific data.
* Deployed on AWS, can be accessed on the web
  + The website and database are deployed on AWS and publicly available, along with deployment instructions. The project is deployable on other cloud providers and the process is very similar.
* Made one time pin feature for update records
  + This feature was implemented to solve the problem of having to manually change record details in the event of a change in consultee information.
  + With the use of SHA-256 hashing we generate a unique one time pin for any record that needs updating.
  + Any third party consultee can then access a separate page which they can enter using this one time pin to submit any changes to their contact details.
  + Any member of the planning inspectorate division can review changes and approve them.
* Added security functionality
  + The API endpoints are secured with CORS.
  + The database access website is secured with Spring Boot security.
  + The one time pin edit system is secure by nature as accessing a record can only be done with a pin.

For continuing with the project, there were some features we would have liked to implement but for various reasons were unable to:

* Adding email verification to the one time pin system
  + For added security, we thought that in addition to the one time pin, a user should have to enter the previous contact email associated with the record in order to edit it. This would make brute force attacks which are already impractical far more difficult
* Adding an approval system for one time pin updates so a database administrator can approve or reject changes.
  + This would further reduce the potential for exploitation of this feature.
* Despite the tool being developed for a small user base, more accessibility features
  + The overall design language we chose for the tool had accessibility in mind, however popup and tooltips in addition to help pages may have made it easier for new users to understand and use the website.