## **InsulatedNarwhal**

Joel Corlett | s1234567}

Mick Hodgson | s3413269@student.rmit.edu.au

Paul McGuire | s3495545@student.rmit.edu.au

# Data Visulization Web App

The Web App will allow users to look at data relating to the council boundaries in the State of Victoria.

Base functionality the user will select a council area from a map of Victoria, then a supplied data source which will then shows the results in tables with accompanied with charts and graphs of the table data.

Enhanced data functionality will allow the user to form their own data base queries to refine the data search whilst also allow the user to investigate more than one council at a time. The user will then have the option to choose the data they wish to see in the graphical output.

Additional data visualization techniques will allow the user to focus on advanced 3D graphic outputs for the council data select.

All data outputs Tables, graphs and charts will have an option to download a report of all information or part information such as graphs or tables only.

Table of Contents

The Team 0

Project Description 0

Core Features 2

Core Feature 1 2

Core Feature 1 Validation Testing: 2

Core Feature 2 3

Core Feature 2 Validation Testing: 3

Core Feature 3 4

Core Feature 3 Validation Testing: 4

Core Feature 4 5

Core Feature 4 Validation Testing: 5

Core Feature 5 6

Core Feature 5 Validation Testing: 6

Core Feature etc.. 7

Core Feature ?? Validation Testing: 7

Extended Features 8

Extended Feature 1 8

Extended Feature 1 Validation Testing: 8

Extended Feature 2 9

Extended Feature 2 Validation Testing: 9

Extended Feature 3 10

Extended Feature 3 Validation Testing: 10

Extended Feature etc.. 11

Extended Feature ?? Validation Testing: 11

Project Estimation 12

Listing Technologies 13

Collaborative workspaces 13

Software 13

Tools 14

Resources 14

# Core Features

## Core Feature 1 – Function web app

Core feature 1 is the simplest part of the project as this mostly revolves around the web app being accessible outside local host and the upload pages / modules and links work together without errors or missing pages or broken links.

1.1 Web app will be hosted and accessible outside the local host. This will be achieved using titan or a third part web hosting service.

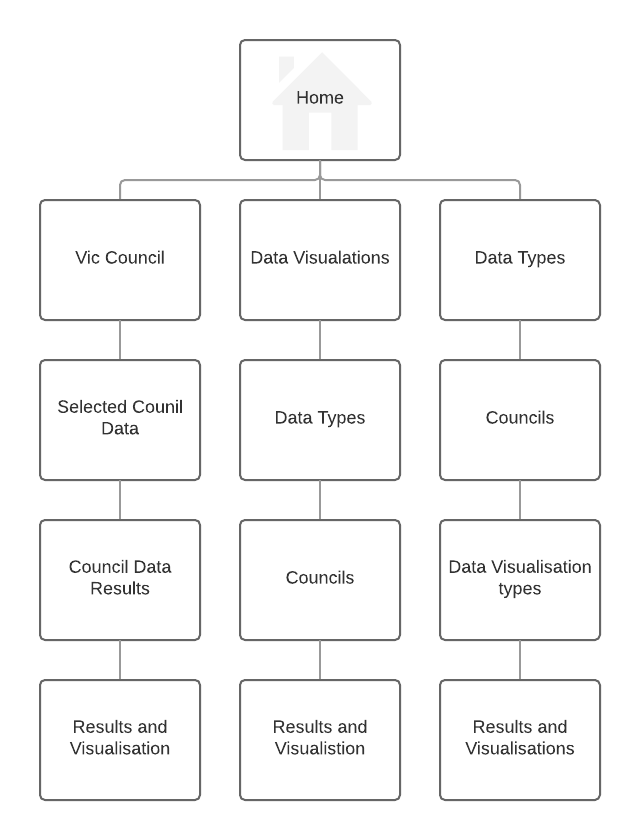
1.2 Web app will include a MYSQL database that will store the council data

1.3 GITHUB will be used to store program revisions so all members can work on the latest version.

1.4 Web app will be developed using XAMPP on local host

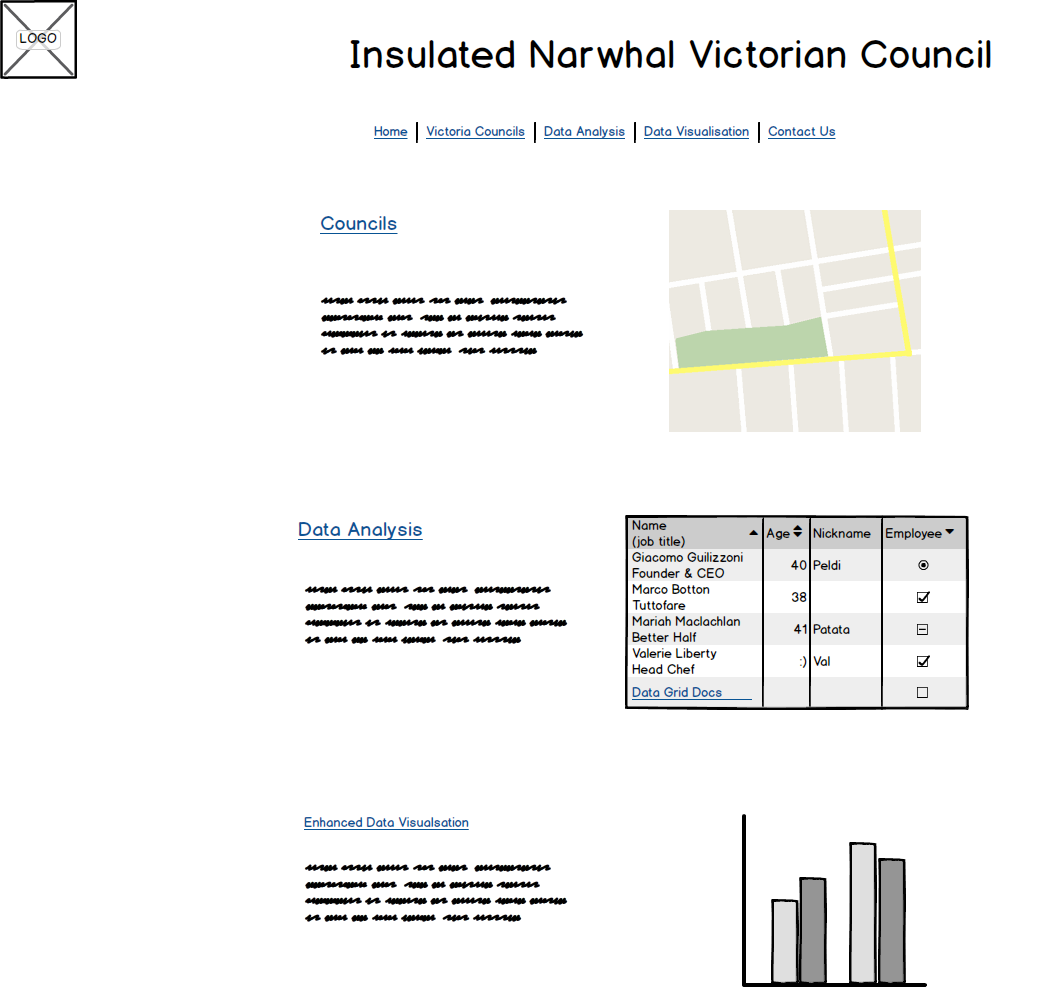
1.5 The operation of the app will start with a home page that contains 3 main links. Basic Data search and Results, Advanced Data search and results, Enhanced data visualization output.

1.6 Each menu option will option to a page that allows the user to select an option the click next updating the page with a new module for the next selection.



Site map from home with 3 main menu options.

WEB APP Home Screen



### Core Feature 1 Validation Testing:

1.1 Users will be able to access the page from the supplied address

1.2 Clicking a link will go to the next/page muddle as interned with no broken links.

1.3 Links when selected will go to the desire page /module as stated.

1.4 MYSQL database will provided for on the hosting service.

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Core Feature 2 – MYSQL database – Victoria Sate councils

2.1 Relational database setup in MYSQL of Victorian Councils, the include data is Council, postcodes, suburbs, median house prices, schools, hospitals. Addition data will be added when source from the site: https://www.data.vic.gov.au

2.2 database structure as follows:

council – council\_id, council\_name

postcodes – postcode, council\_id

locality – suburb, postcode, property\_count

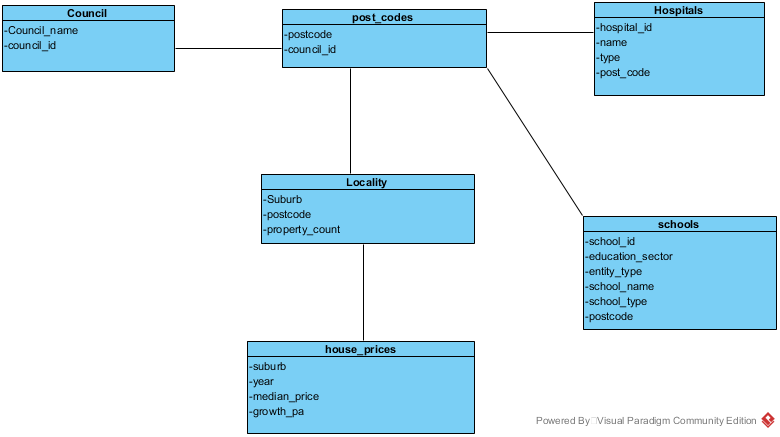
house\_prices -suburb, year median price growth\_pa

hospitals – hospital\_id, type, postcode

schools – school\_id, education\_sector, entity\_type, school\_name, school\_type, postcode

Copy of the data file database\_data.xlxs for upload into MYSQL database





**MYSQL Database Diagram showing relationships**

### Core Feature 2 Validation Testing:

2.1 The file datasbe\_data.xlxs can be uploaded into the MYSQL database once created.

2.2 MYSQL database fields match the data upload

2.3 Can perform SQL queries on the created database via a Database GUI

2.4 can perform the required WEB APP SQL queries with correct results.

2.5 Can communicate with the database via web code (PHP, JavaScript)

2.6 The database can be uploaded / installed into web hosting service for use.

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Core Feature 3 – Council Selection.

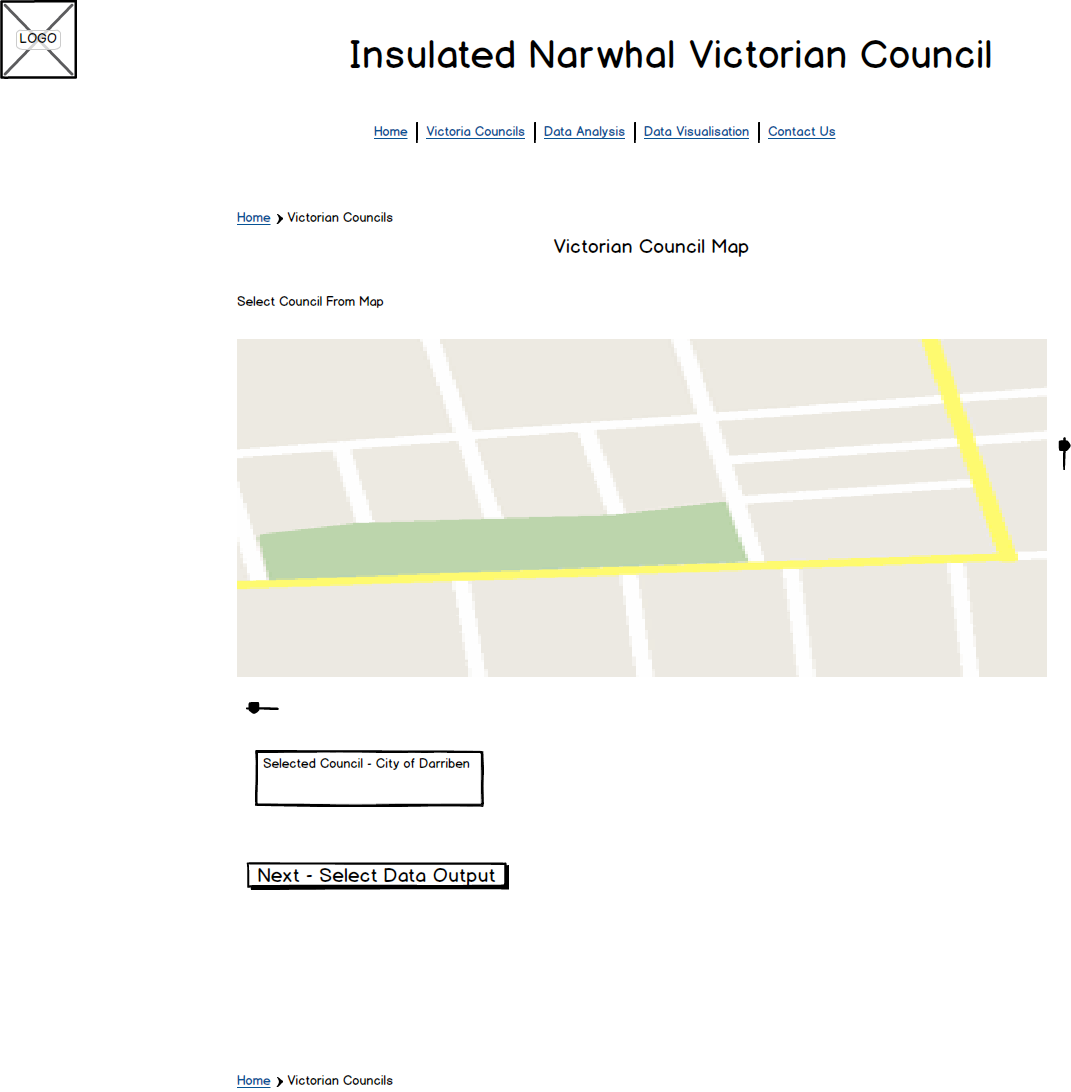
3.1 User will be able to select a council from the map of Victoria.

3.2 When user hovers over a council area the council name is displayed in a small popup window

3.3 Selected council is highlighted on the map.

3.4 Selected council name is displayed.

Map Selection screen.



### Core Feature 3 Validation Testing:

3.1 User can navigate to the council selection screen.

3.2 User can hover over map and see the name of a council in a popup window

3.3 User can select the map and once selected the council area will highlight and the select council name displayed.

3.4 User can unselect a council by clicking on the council area in the map again.

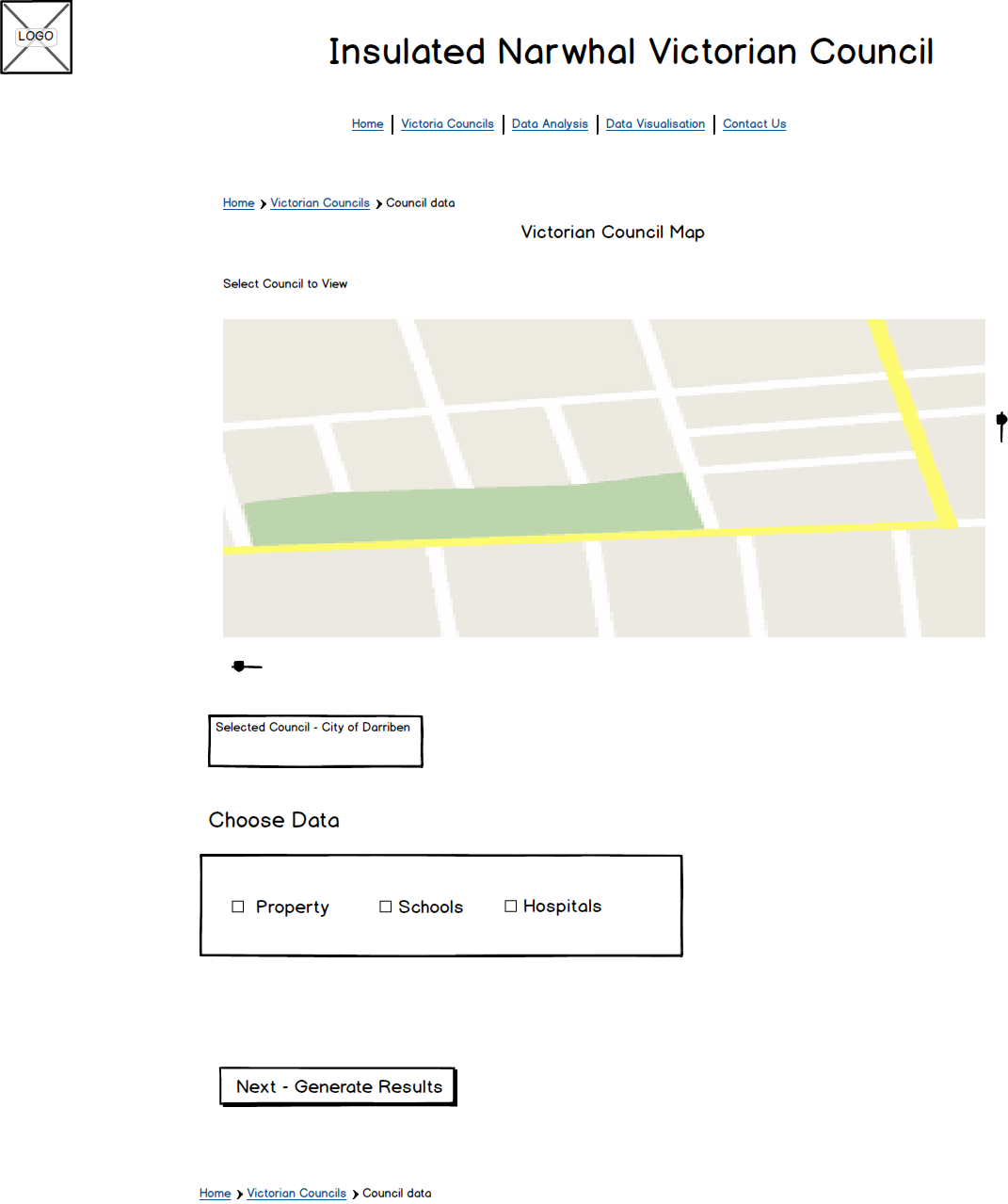
3.5 User can select a new council by clicking on it in the council map. If a council is already selected user will be notified via pop message they are changing councils.

3.6 User can click next to go to the next screen in the process Data Selection.

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Core Feature 4 – Basic Data Selection

Data Selection screen opens under map Selection. This feature is one of the more difficult and important as it involves connecting to the vic council database, making sql queries to retrieve data then displaying the retrieved data in tables.



4.1 User will be able to select one of 3 data sources House Price, Schools, Hospitals.

4.2 User will then click next to retrieve the data. At which point the web app will display the selected council, associated postcodes, suburbs and the data pertaining to each in tables.

### Core Feature 4 Validation Testing:

4.1 Data can be selected via radio button

4.2 Only one data source can be selected.

4.3 Data source can only be selected by clicking the radio button

4.4 User will only be able to move next section with selected data.

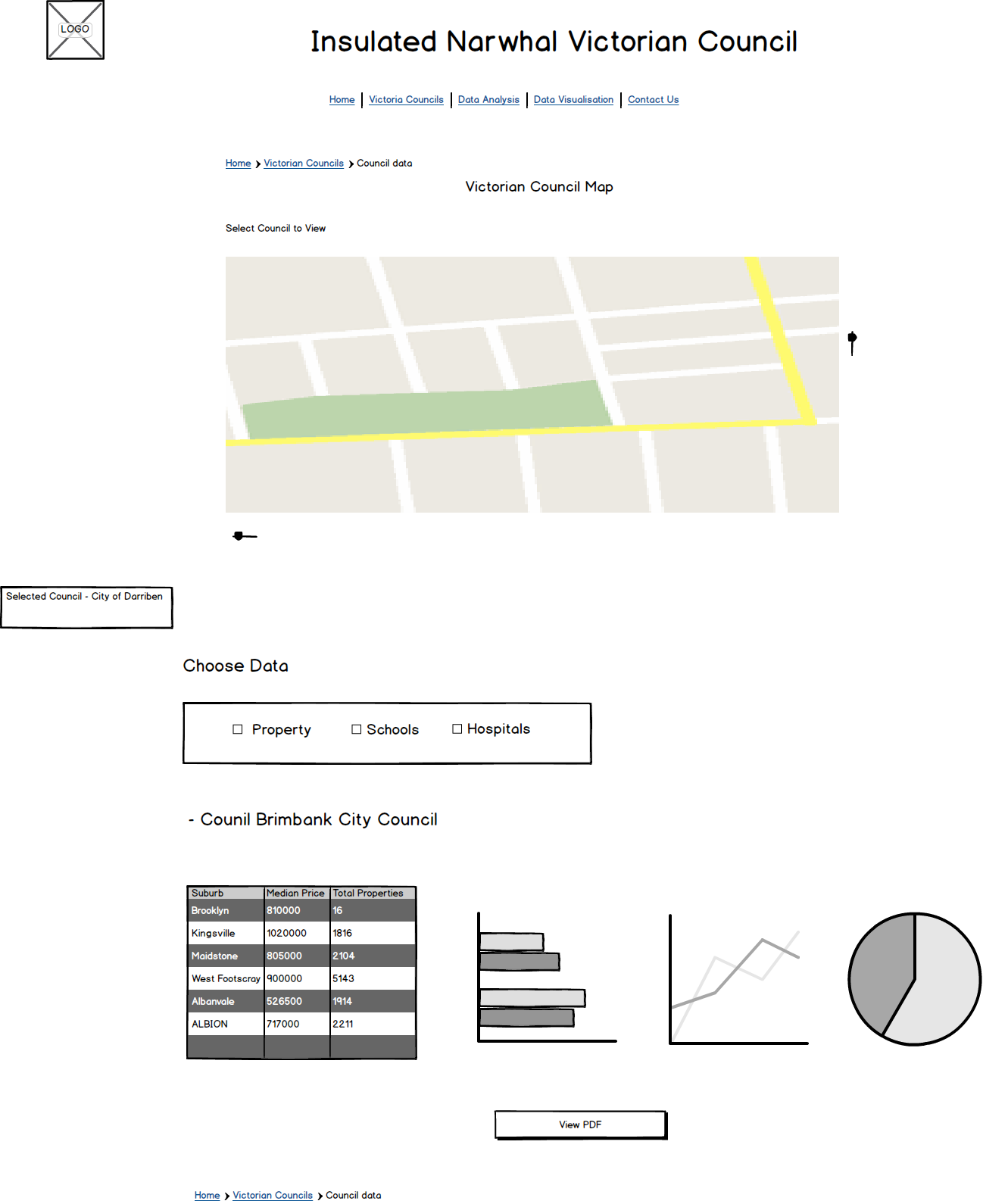
4.5 Selection will access the database and select the correct council and data information.

4.6 the built in query will select all council information for the required postcodes and suburbs outputted to tables. Results will be correct with no errors or omissions.

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Core Feature 5 – Data Results

Once the user clicks next the data results will be displayed in tables for all postcodes and suburbs contained in the council with the data for each postcode and council.



5.1 Two Tables will be displayed one for post codes within the council and one for Suburbs.

5.2 Each table will display the data in columns with each row representing a unique postcode or suburb

5.3 Graph images are shown that users can click on to select for output. Graphs types listed will be Bar (vertical or horizontal), Line and Pie Chart

5.4 User can click a graph type to select for output

### Core Feature 5 Validation Testing:

5.1 Table output will be correct for the council and data source selected.

5.2 The Graph icons will be displayed and hover will indicate what graph can be chosen.

5.3 Graph icons can be selected for inclusion in output by clicking on them

5.4 Graph icons can be remove from Selection by Clicking on them when selected.

5.5 Click View data can only be achieved when at least one graph type has been selected.

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Core Feature 6 View Print Data PDF

User will be able to select Print data for a pop up window that will show the data tables produced plus selected graphs. User will then be able to save the information to PDF or send to printer.



{Replace Image with your design requirements}

### Core Feature 6 Validation Testing:

6.1 After user has click Print Download button window will pop up.

6.2 User will be able to close window without saving or printing

6.3 User can click save and be asked to enter location file name in save window

6.4 user can then click printer and local printer can be selected in print window

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

# Project Estimation

{Include estimation spreadsheet and justify}

# Listing Technologies

## Tools, Resources

XAMPP – desktop PC WEB server

HTML PHP CCS JavaScript editor (Eclipse)

Excel – data formatting

MYSQL – Data base creation

MYSQL Dashboard – GUI for database creation

[https://www.data.vic.gov.au](https://www.data.vic.gov.au/) for data sources

[https://www.w3schools.com](https://www.w3schools.com/) for WEB programming tutorials

<https://developers.google.com/chart/> for Data visualization tutorials, examples

[https://d3js.org](https://d3js.org/) for JavaScript Data visualization examples tutorials

<https://support.google.com/fusiontables/answer/2527132?hl=en> Fusion tables for map creation

## 

## 

## 

## 

## 

## 

APPENDIX

# Extended Features

## Extended Feature 1



{Replace Image with your design requirements}

### Extended Feature 1 Validation Testing:

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Extended Feature 2



{Replace Image with your design requirements}

### Extended Feature 2 Validation Testing:

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Extended Feature 3



{Replace Image with your design requirements}

### Extended Feature 3 Validation Testing:

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }

## Extended Feature etc..

# 

{Replace Image with your design requirements}

### Extended Feature ?? Validation Testing:

**Author:** { team member / designer } **Create Date:** { dd/mm/yyyy }