FIT31612-FIT3164 Semester 2 2023 User Guides

13/10/2023

Project:

Analysing the rising prices of Australian grocery products (P5)

Team S2_CS_03:

Marcus Fox	(30608252)
Minjih Kim	(31475329)
Carter Chin	(32418361)
Putu Sari	(31943470)

Table of contents

Part 1: End User Guide

Accessing the software

Since our website is not currently hosted on a server, we'll run it locally on our machine. To start the web application, please follow the technical guide below.

Navigate to the Project Directory: Change your working directory to the project folder.

Install Dependencies: Use npm (Node Package Manager) to install the project's dependencies: npm install

Start the Development Server: Run the following command to start the local development server: npm run dev

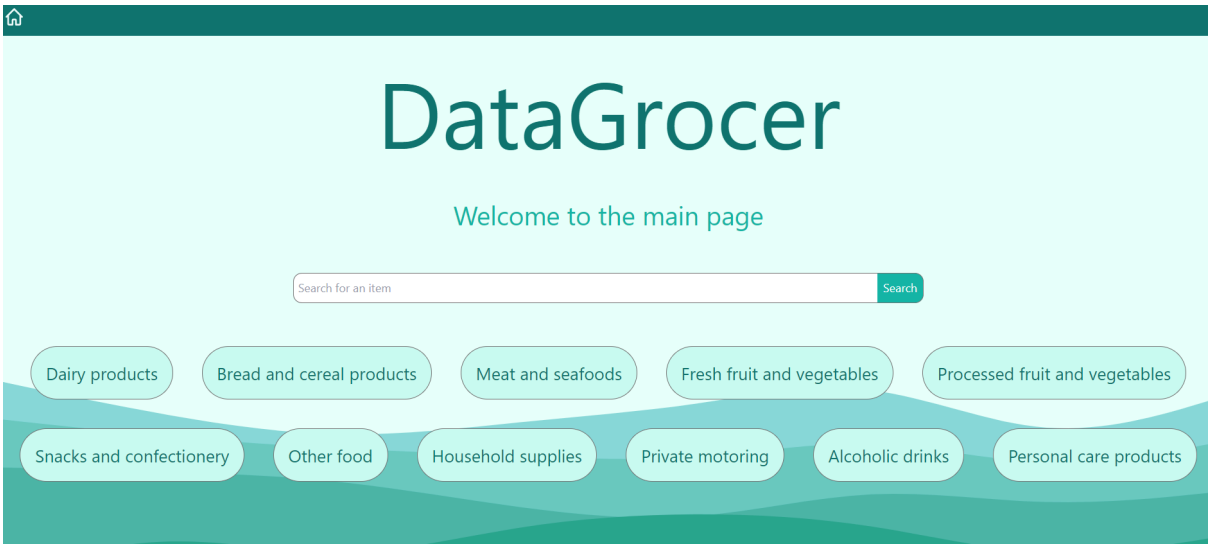
This command will build and run the application on your local machine.

Access the Application: Open your web browser and navigate to http://localhost:5173 which will be the main page.

By following these steps, you'll be able to run and access our web application on your local machine.

Searching for product

This is the website's main page, where users will have two methods to search for items. You can either find a specific product by entering its name in the search bar, or explore items by their categories which are the buttons below the search bar.

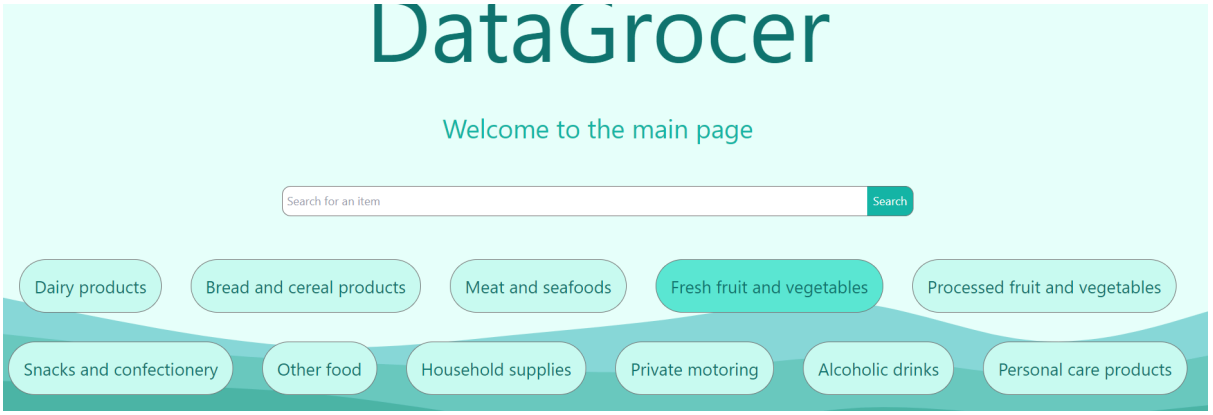


For instance, if you search for "milk," you'll be directed to a dedicated page displaying related products, such as regular milk and chocolate milk. By selecting an item, you'll be

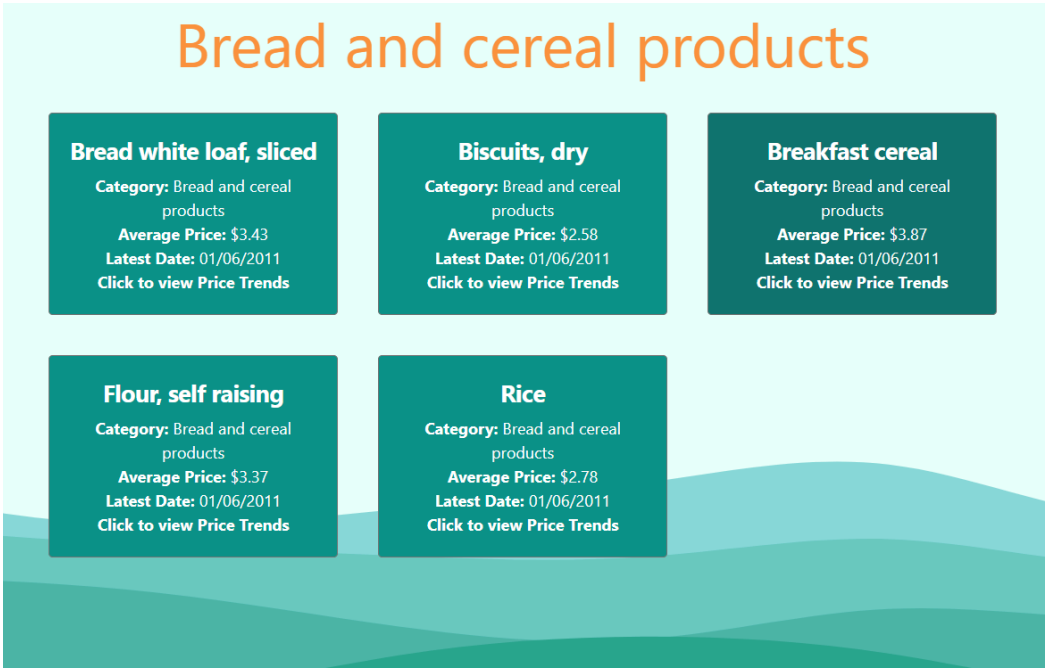
directed to its respective product page, where you can access pricing details and more information. If the entered item name does not match any of the items we have then it will let the user know that there are no results.

Browsing categories

The other method of finding an item is through clicking on one of the categories that the user would like to browse, for example, If the user wants to see what fruit data we have then they can select the Fresh fruit and vegetable category which will direct them to the category page.



On the category page, you will find a collection of related items, each showcased using a Card component. These Cards provide more than just the product name; they also display the average price data across Australia based on our latest information providing you with essential information as you browse through the products in the category. Clicking on the component will redirect you to the item page.

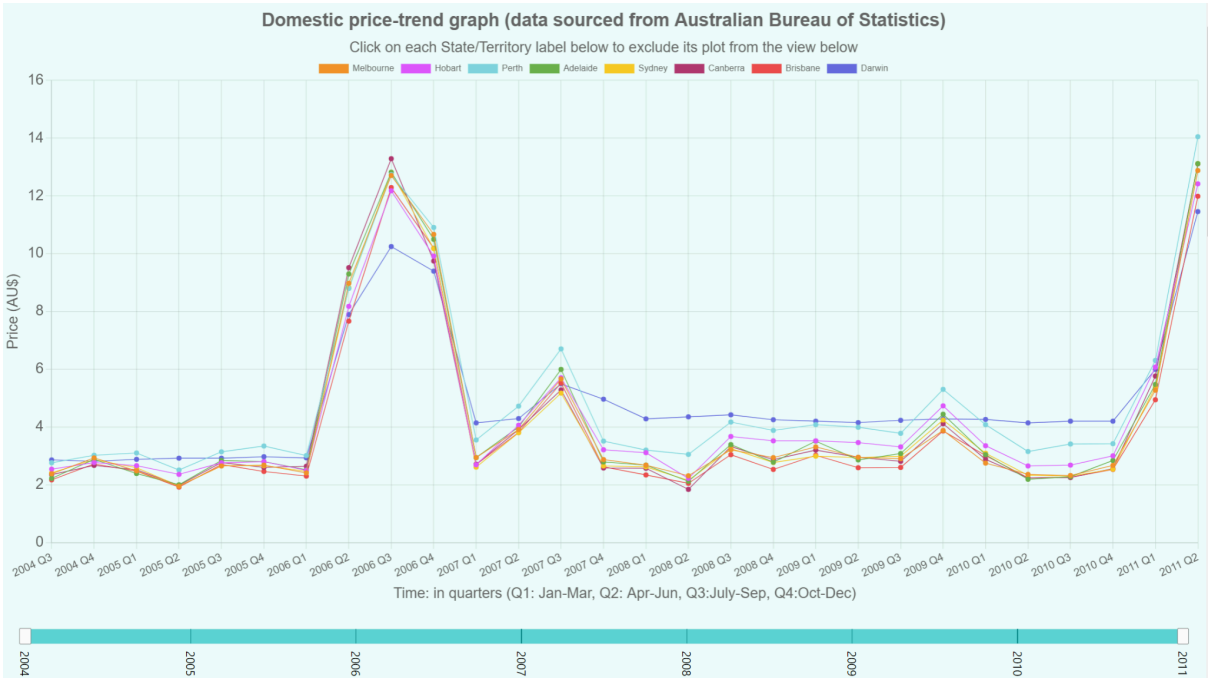


Viewing price trends

Once the user lands on the individual item or product page, they will be able to access three different data visualisations that describe the trend of the price for that item, over time as stored in the database.

1. Main price trend graph

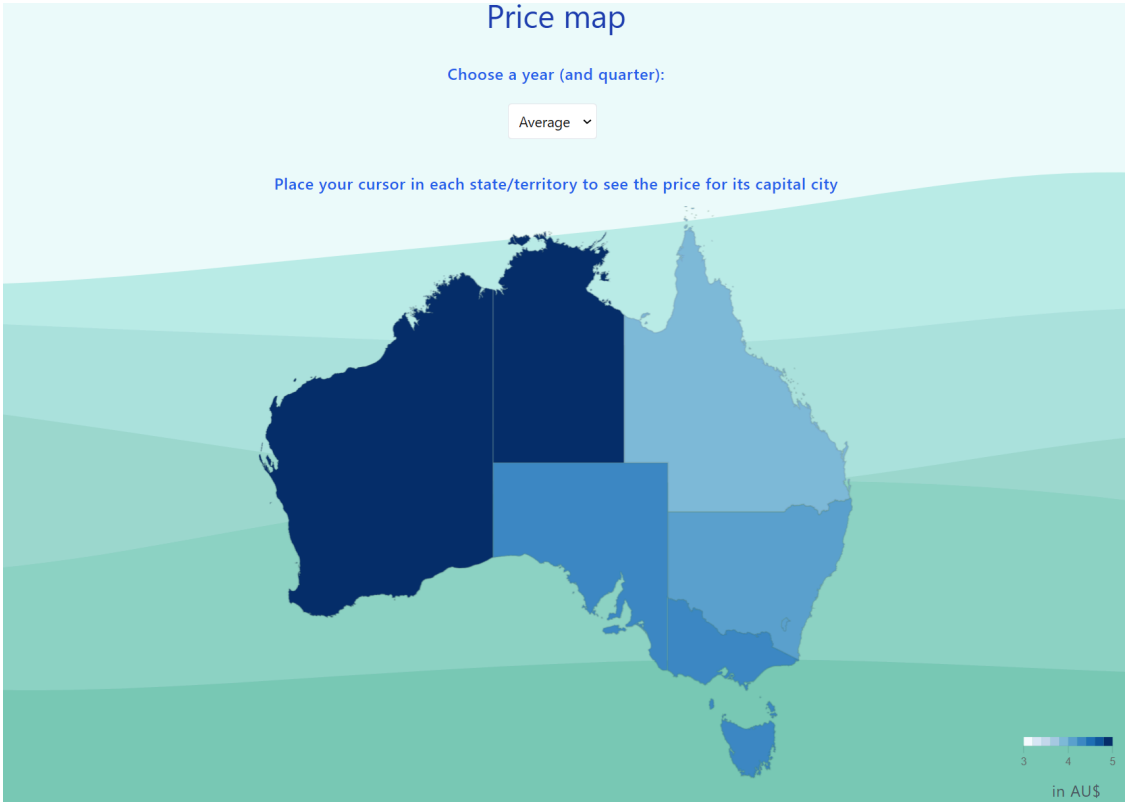
The first graph that appears on the page is the line graph for the item specified that plots its price over time; the time is recorded in quarters as it was originally in the data obtained from the Australian Bureau of Statistics (ABS). The axis labels (i.e. their units) are clarified in the title for each axis.



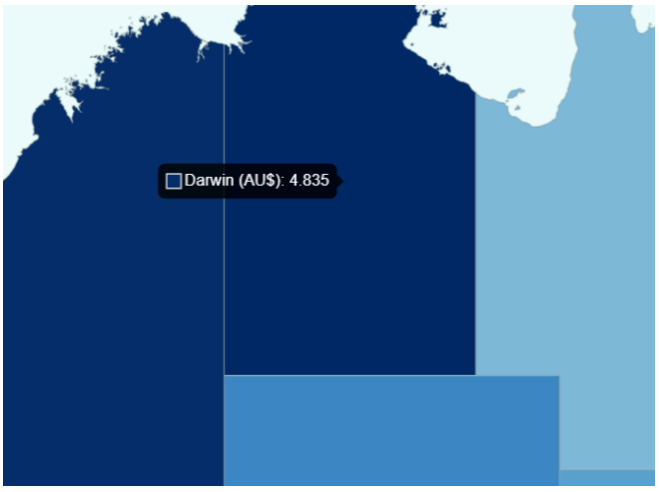
All lines are colour-coded to distinguish them from each other; each line represents a capital city of Australian State/Territory such that the user can view the difference in price trend across different cities for the product of interest. The number of data points (i.e. the time range) displayed may be different for each product as it depends on the data originally provided by ABS.

2. Price map

Scrolling down the page, the user will see a map below the 1. Main price trend graph.



By default, the map displays the average price (across all years available for that item's data) for each capital city. As hinted by the instruction displayed above the map, when the user places the cursor on each State/Territory, a label will pop up (see the screen capture below) that shows which city they are looking at and the (average) price of that item for that city.



The map is colour-coded such that it follows the colour gradient scale that reflects the price scale for that item. The darker the colour, the higher the price. The guide to customising the map is provided in the next section: Customising data visualisation.

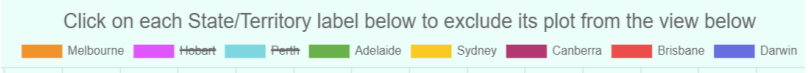
3. Comparison graph

Scrolling back to the top of the page, we revisit the line graph because it can be used to create a comparison graph: a graph that can compare the current page's item with any other item in the database. The user is encouraged to interact with the application to add a second plot to the graph, to compare the price trends of 2 different items. Instructions on how to interact with the features are provided in the following section.

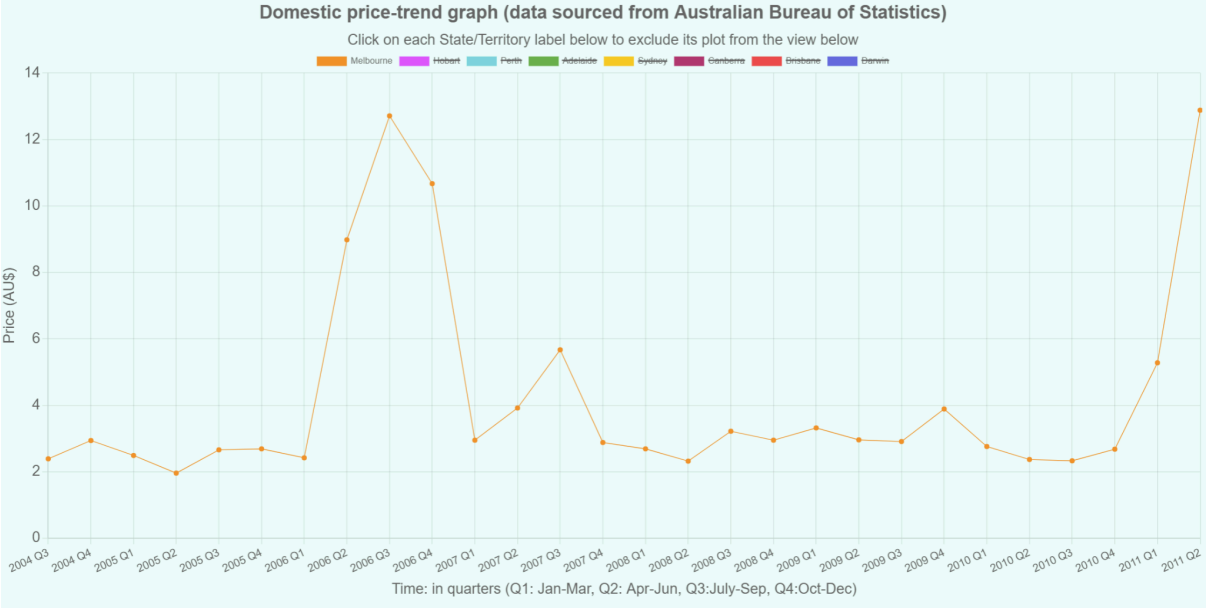
Customising data visualisation

1. Main price trend graph

The user can customise this graph in two main ways. First, they can choose which data (i.e. which city) to show on the screen by using the colour legend provided on top of the graph. Each label acts like a button, and when the user clicks the label (either the text or the colour block), the corresponding city plot disappears from the view. When the click event is successfully registered, the labelling text is crossed-out like below (see Hobart and Perth):



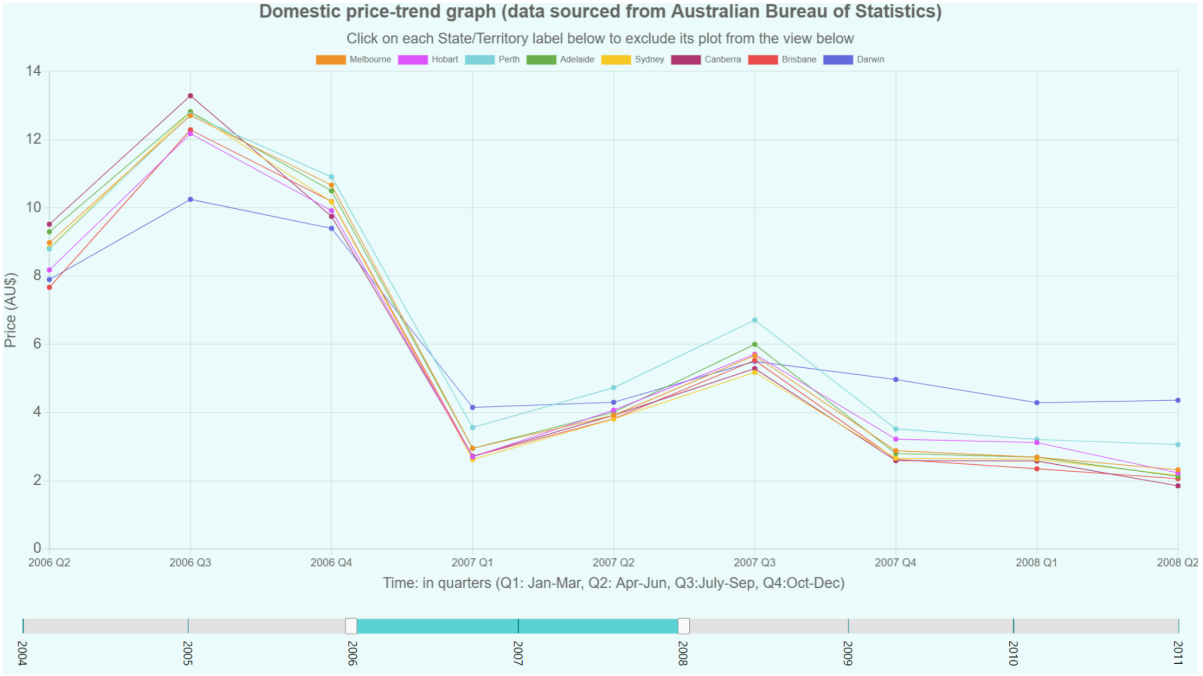
This indicates to the user that those cities have been removed from the graph. They can click again to add them back to the graph. For instance, the user can create a price trend graph for Melbourne only, by excluding all other cities:



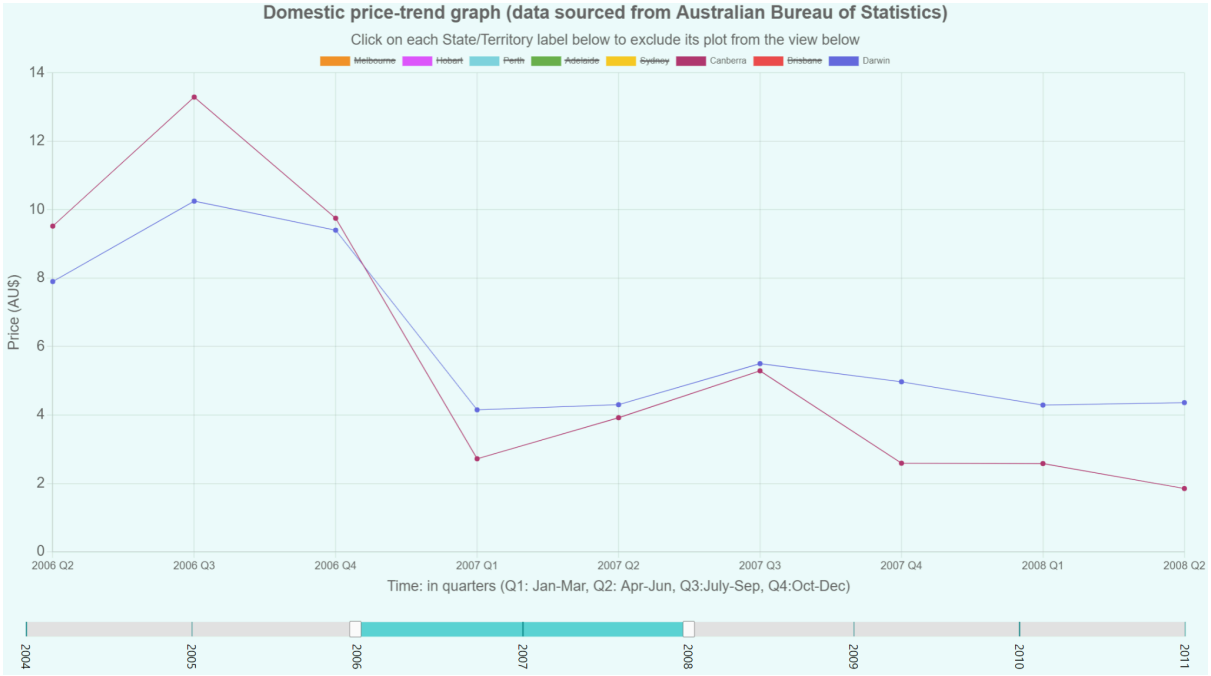
The original/default state can be recovered by either 'un-crossing' all the buttons, or simply by refreshing the page.

The second way to customise this graph is by making use of the slider. They can click and drag the white button on the slider (left and right) to adjust the time range the data is shown

for. By default, all the data available for this item in the connected database is shown to the user. However, using the slider, the user can customise the graph to show the plots starting at an earlier time and/or ending at a later time. For example, instead of 2004 Q3 to 2011 Q2, the user can see the data for 2006 Q2 to 2008 Q2 like shown below:



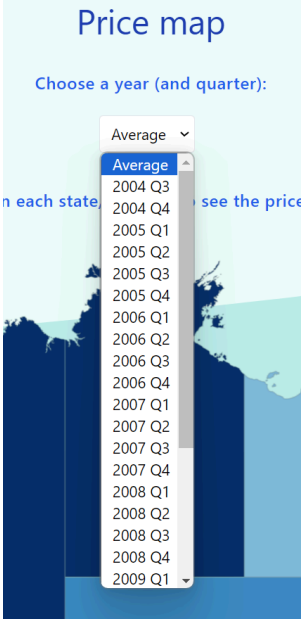
From the current state, the user can also customise the number of plots shown by following the instructions provided earlier (i.e. clicking/unclicking appropriate city labels in the legend). For example, they can choose to view the data for Darwin and Canberra for the time range selected above:



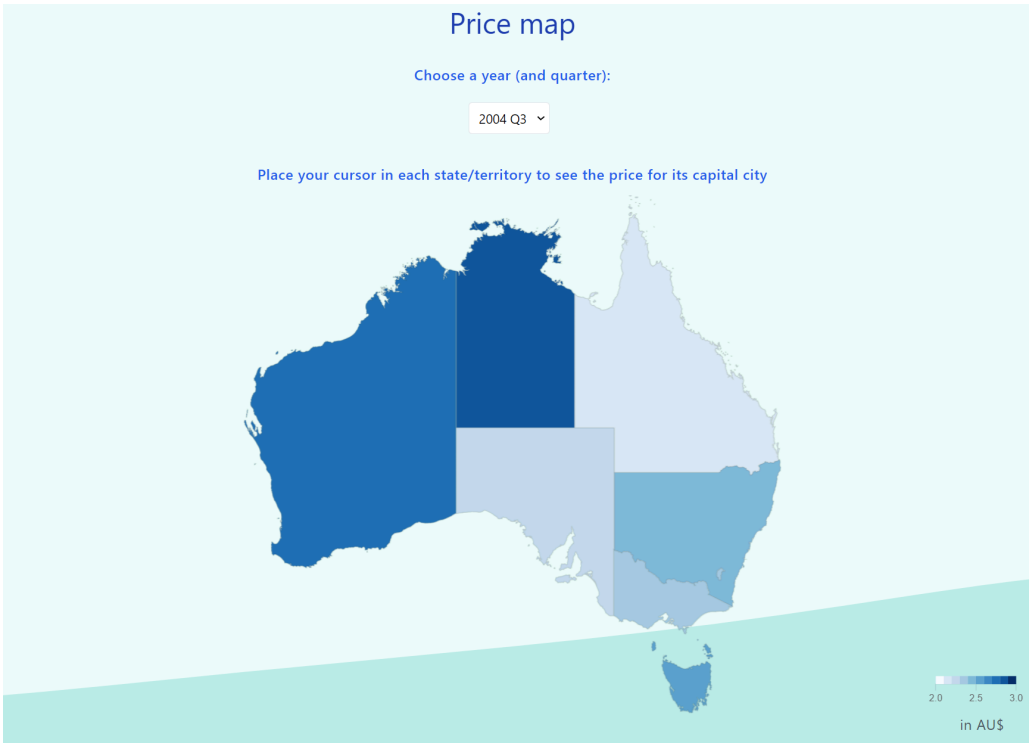
To revert back to the original state, the user can drag the slide limiters back to the end of the scale, or they can refresh the page.

2. Price map

The price map can be customised by choosing which year and quarter they want to view the data for. When the user clicks anywhere inside the “Select” drop-down button, a list of years and quarters will be shown; same as the horizontal axis of the line graph. The user can scroll through to see all available options.



When the user chooses a time option from the drop down, the map will change immediately, and show the price map for the chosen year and quarter.

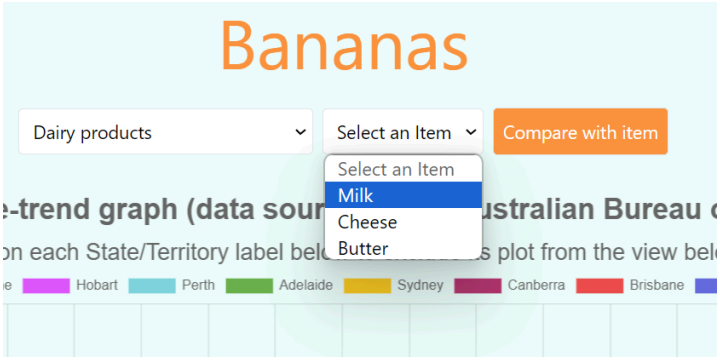


The user can tell that the map has been updated from the colour change: the colour scale will change year to year and quarter to quarter since prices change throughout time. To view the average price map again, the user can reselect the ‘Average’ option from the drop-down or refresh the page.

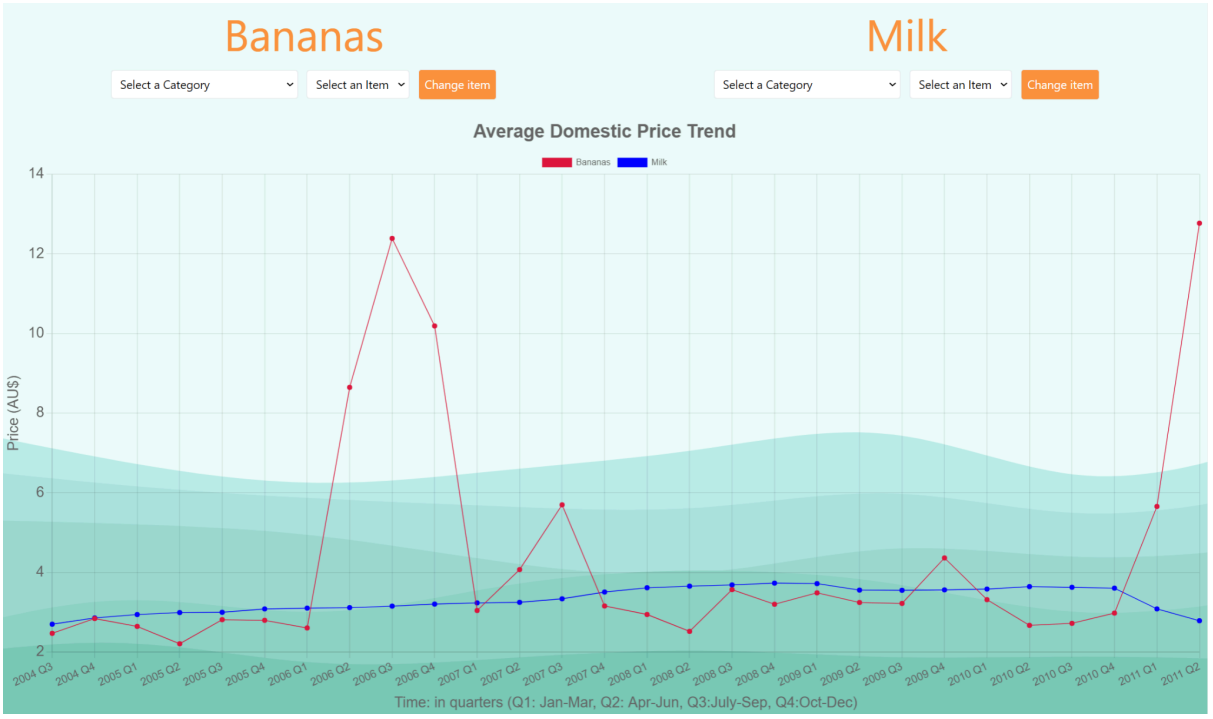
3. Comparison graph

The comparison graph can be customised as follows. The user needs to select which product they want to compare the current product with. To do this, they first need to select which category they are interested in for the comparison.

Once they have selected a category and when they open the drop-down for “Select an Item” next to it, they will be provided with all items they can choose from based on their category selection. For example, if they choose the category “Dairy products” they will see the following options for the item:



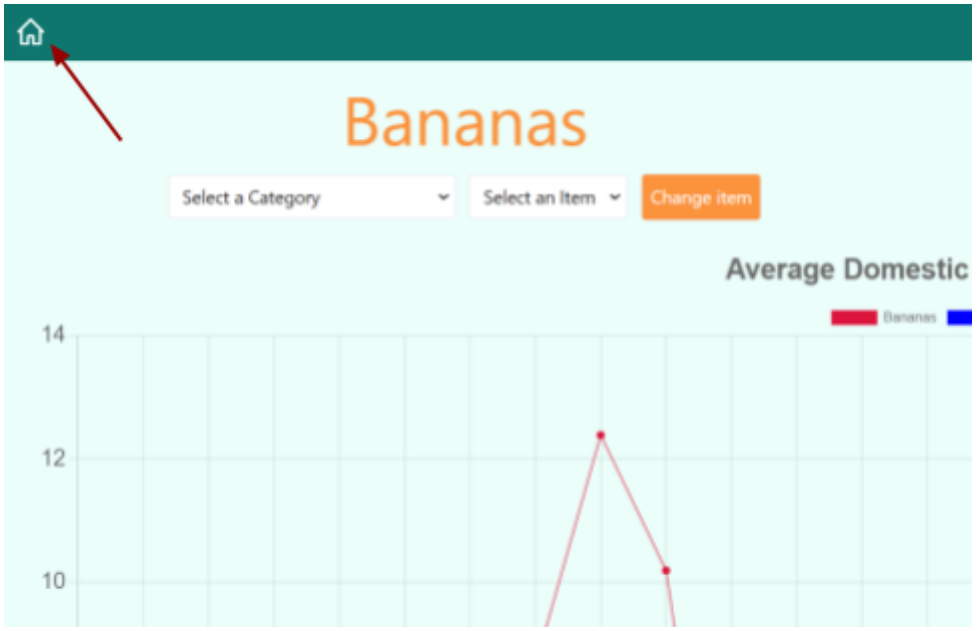
Once they make a selection and press the “Compare with item” button, the user will be redirected to a new page and they will see an updated version of the line graph from the previous page, with the selected item average plot added in blue colour as indicated by the legend.



In this way, they can compare the average price trends of two products on the same set of axes, across different categories as well. They can compare only 2 items at a time, this was a design choice to avoid creating a cluttered graph that will be hard to read.

Exiting the software

Exiting the software is simple. To go back to the main home page, the home icon is available at the top left corner of the page any time.



To exit from the website entirely, the user can simply close the browser.

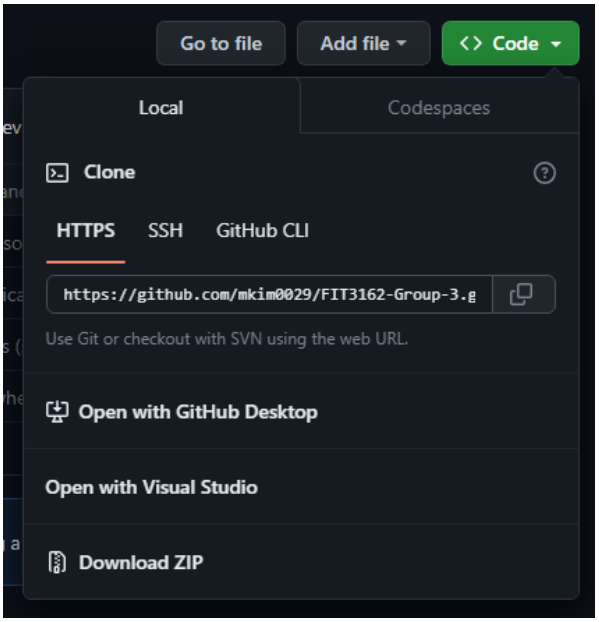
Part 2: Technical Guide

All instructions in this technical guide pertain to installation on a Windows based system.

Getting access to application files

The application files are hosted on Github at the following link:
https://github.com/mkim0029/FIT3162-Group-3

To gain access, you must have a valid Github account and request access from mkim0029.
For a quick installation, you can simply download the files by clicking Code -> Download ZIP



The zip can then be extracted to a working directory. It contains three important folders:

- App: frontend web server
- Database: files for setting up and importing data into the database
- Data: datasets that are used by the application

Setting up the database

1. Download the PostgreSQL installer

The installer can be obtained by downloading it from:
https://www.postgresql.org/download/windows/

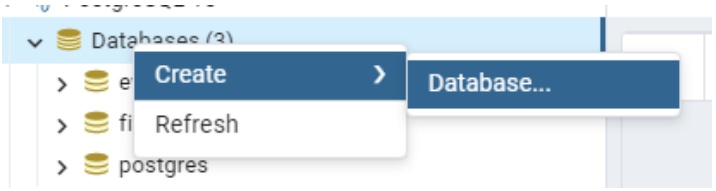
2. Install PostgreSQL

Run the installer and continue with all the default options. Ensure that you set a secure password for the ‘postgres’ user during installation and make note of this password as it will be required later for the frontend.
The setup can be confirmed by checking that ‘pgAdmin’ is now installed and can be opened.

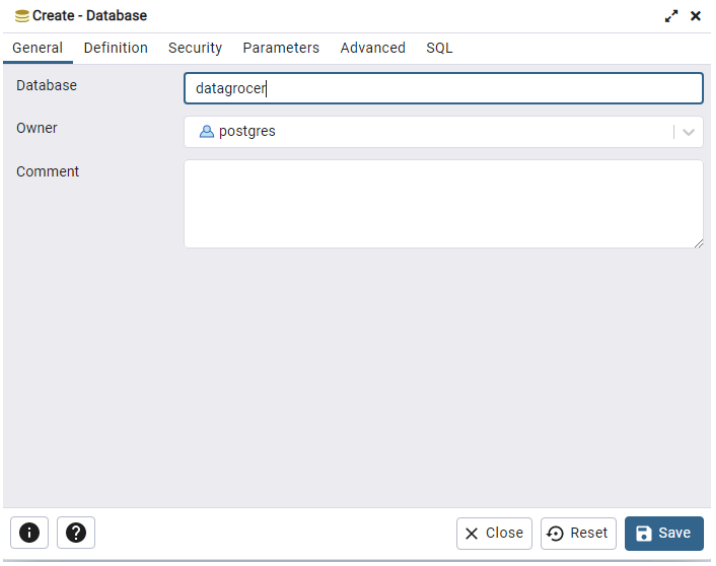
Adding data to database

1. Create new database

Launch the ‘pgAdmin’ tool, a dialog box should appear and prompt you for the password to the ‘postgres’ user you set during the installation.
Create a new database by right clicking on Databases and navigating to Create -> Database…

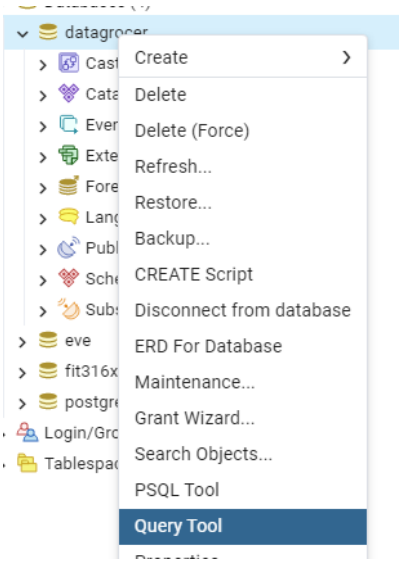


Now we can enter a name for the database, here we have chosen ‘datagrocer’.



2. Create database schema

We will now run the schema to set up the structure of the database. Launch the ‘Query Tool’ by right clicking on the database we just created.



Open database/1-create_schema_and_tables.psql file by clicking the ‘Open File’ button and then execute the schema by pressing F5.

3. Import data into the database

We will be importing data using the provided datasets and the database/import_csv.py file. Python must be installed by downloading the installer for the latest version from:
https://www.python.org/downloads/windows/
Once python is installed, we will need to install some dependencies for the importing script. This can be achieved by running ‘pip install pandas sqlalchemy’.

To ensure the script connects to the correct database, we must edit the connection string in the script.
The connection string is located at the end of database/import_csv.py and by default looks like ‘ conn_str = "postgresql://postgres:password@localhost:5432/fit316x" ‘.
In this line, change the password, host, and database to the correct values.

The password will be the one that was set during installation. The host will be the host of the database, if you are performing this locally it will be ‘localhost’. The database will be the name you created during the database creation step, for this guide we would have to replace it with ‘datagrocer’.

Finally we can run the ‘database/batchimport.bat’ file to iterate over all the data files in data/clean/ and import them into the database.

Setting up frontend server

1. Installing Node and NPM

The web server requires Node and npm to run correctly. These should be installed using the installer downloaded from the official node website:
https://nodejs.org/en

2. Editing the database connection

Similar to how we edited the connection string in the python script, we must also set the correct database for the web server.
Navigate to app/src/lib/index.js and at the top there will be a Knex object.

```
const knex = new Knex({
  client: "pg",
  connection: {
    host: "fit316x.cvsli88352ui.ap-southeast-2.rds.amazonaws.com",
    database: "fit316x",
    password: "password",
    port: 5432,
    searchPath: "absdata",
    user: "postgres",
    ssl: {
      sa: "mysql-ssl-ca-cert.pem",
      rejectUnauthorized: false,
    },
  },
});
```

Here we must once again set the correct password, host, and database. Use the same values you used for importing the data into the database. If the server is being run locally, the ‘ssl’ argument can be removed as a SSL connection won’t be required.

3. Starting the web server

Navigate back to the app directory and open a command prompt here.
The dependencies for the web server can be installed by running ‘npm install’.
Once that is complete, the application can be started by running ‘npm run dev’.
This should print out a web address in the command prompt that can be navigated to in the web browser.