**CAB 230 Client-Side Report**

**Adam Conway n9819801**

Introduction

This report will cover the client-side functionality for the cab 230 project. This single page web application purpose is to serve data to the user about crime statistic in multiple formats. The features implemented in this projected are listed below:

Adding Users (signing up),

Logging Users in,

Show Offences List,

Show Offence LGA and total in table,

Refine data in table by, Age, Gender, Area and Year,

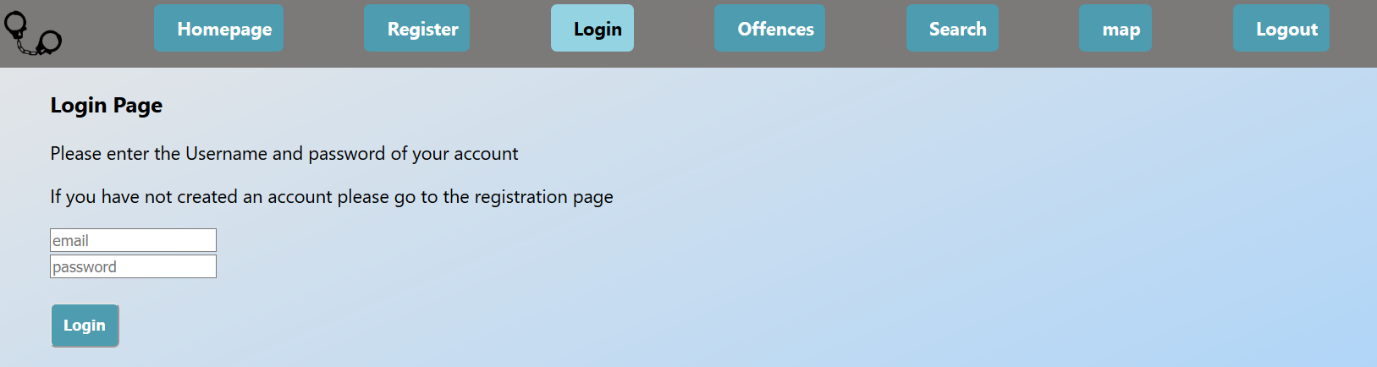
Show data on map with markers on LGA.

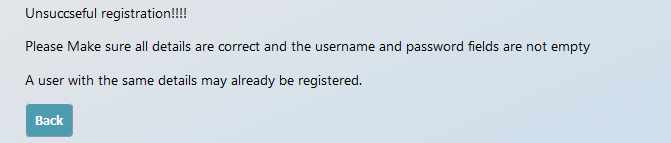
Graphing functionality was not implemented in the project and sorting the table by the headers was not implemented. Explanations will be given later in the report. The goal for me in this project was to implement as many working features in this project without the help of outside libraries where possible. This meant styling for the project was to be written without the help of libraries such as Bootstrap and Material-UI.

Features

The navigation bar is the main component of the web application as this is what controls what information is loaded onto the page. All the functionalities are accessed by clicking the buttons on the navigation bar.

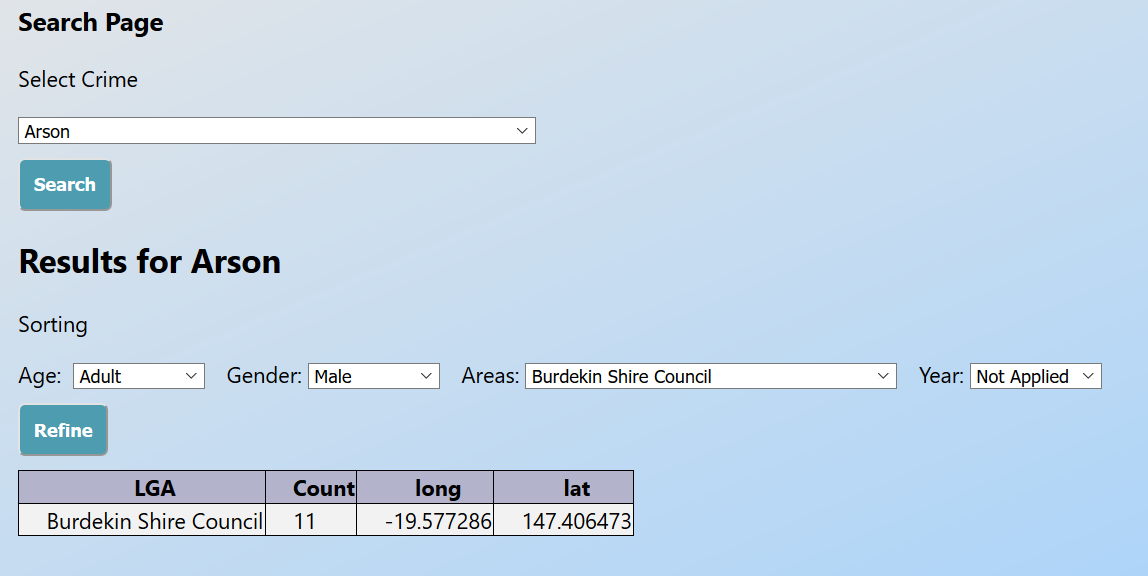
The Home page is where the user is first taken when the webpage is loaded, this page contains information what functionality is available for the website and what the user should do next.

The Register, login and logout page all have similar formats with a simple title and a form for the user’s email and password.

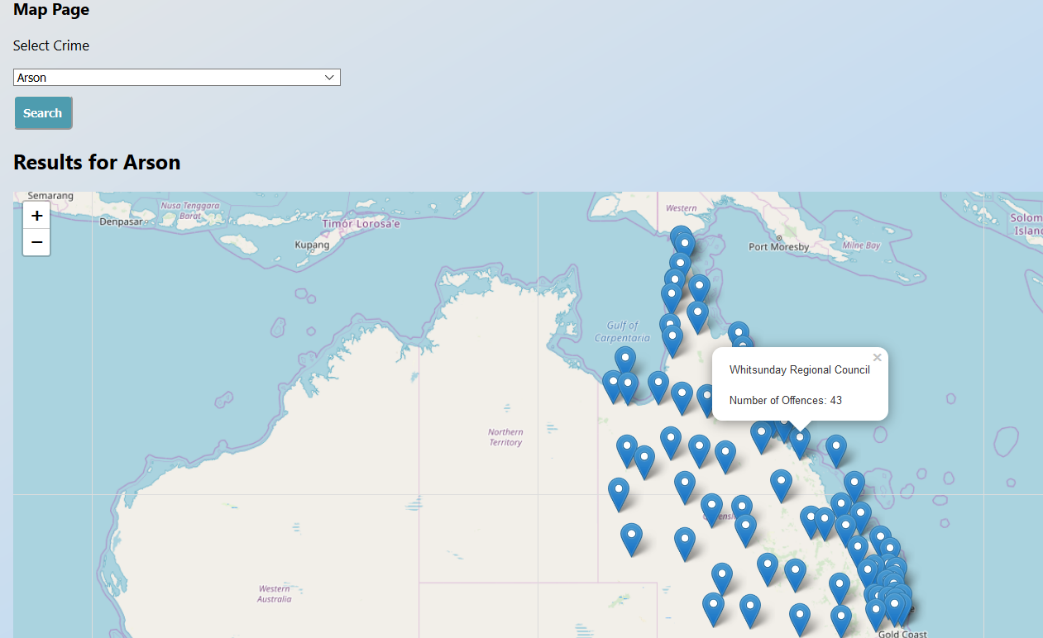
For both pages once the button is pressed the page runs checks the values in the forms and returns a response depending on those values. If the account details already exist for registration the user is told. Similarly, to logging in if the details are incorrect the user is told. If any invalid characters are used in the forms the user is told not to use them as this may cause errors.

The offences page shows all currently registered offences in a list, this page has little functionality for the user.

The Search Page is the contains the most functionality, the page requires the user to be logged in. The page first displays a title with a selector box filled with the offence list. Once the user selects the crime and clicks on the search button the page then loads in sorting functionalities for the results. A table is then generated with all the results for the selected crime. The user then can refine the results using the given items.



The map page, like the search page requires the user to be logged in. The page first load in like the Search page with only a selection box and a button, once the button is pressed and page finished loading a map is shown with markers showing the LGA binded to popups with the reported total number of the selected offence in the area.



Technical Description

Registration

Registration uses a form to collect information about the user. If When the Register button is clicked, the information in the form is collected and checked to see if any invalid characters are used. If the check passes a request to the server is made to see if the detail currently matched one already stored in the system. Once this passes the details are stored on the server and the *username* and *password* are stored as global variables to be accessed later. I didn’t want to use global variables and tried to use a parent state that passes on values to the child components. This part worked, but I could not understand how to pass the state back to the parent or the child’s siblings (aka Login). Once the registration is valid the webapp automatically directs to the login page.

Login

Login also is a form page but has a different purpose. Before login loads the form onto the screen it checks the global variables to see if they are empty. If *username* and *password* are empty the page loads the form with placeholders in the boxes. If the *username* and *password* variables are not empty the boxes are prefilled with the username and password of the user. The final global variable used is the *usertoken*, if the user token is null the page is loaded depending on the *username* and *password* variables. If the user token is not null the page loads with a message that a user is currently logged in and the user is directed to log out of the current account to login. Logging out just sets the *usertoken* to null and takes the user back to the login page. If the user successfully logs in they will be taken to the offences page.

Search

The search page is the most complicated page of the application. It has the most functions and also is the page that makes the most request to the server. The page starts with just a selector box and search button. The selector box Is filled with the data retrieved from hitting the offences endpoint. Once a user selects the crime they want to see statistics for, multiple request to the server are made. The first request is retrieving the data for the table, this request has to be made with the *usertoken* to be successful. Once the data is retrieved more request to the server are made. These requests are used to get the offences, Areas, and Years to fill the new selectors with. I could have made request for Age and Gender as well but as those are limited in option hardcoding works for now. If there were plans to increase the range in ages and genders to search by, I would have fetched them from the server. Once the selectors are filled the page is loaded with the new selectors available above the requested data for the selected crime. The user can now refine the search by making selections in the selectors and hitting the refined button. This will make another request to the server but with the added refinements selected.

There were many problems in this page during development but most of them stemmed from not understanding how to make request to the server properly. The were other issues which caused problems, but some were fixed. The first was the feedback to the user on what they had searched and refined data for. When the request to the server is made the page goes into a loading state when its finding data, when the loading state is finished the data is shown but the selectors were all reset back to the default setting so if the user could not remember what they searched for the data retrieved did not make sense. I tried to save the selected index of the selectors and put that as the default, but this created problems of overwriting the not applied index along with doubling up on the selected index. The easier solution was to allow the selectors to reset but have a note showing what the users crime selected was along with a tag showing what the user refined the data by.

One issue I was not able to solve was sorting the data by headers. As I did not use React Bootstrap Styles, I had to make my own table sorting function. I had previously used one on another project that works on pure html tables but due to the way React requires tables to be written this function did not work. I tried many different solutions but came to the conclusion that React tables was the only way to do it. I then tried re-implementing the tables but did not understand how to get the data into the required format and return it properly, so I just returned to my old method and left it uncompleted.

Map

The final technical part of this project that was completed was the mapping functionality. Instead of making one from new I decided it would be best to use the leaflet library along with its React handler. Similarly, to the search page the map page will not load if the *usertoken* is empty. The page is loaded with a selector box with the offences list. Once the search button is pressed the page is set to the loading screen as the map is prepared with the date. A request to the server is made to retrieve the data about the offence and the leaflet map is then created with its starting position set to show QLD. Markers are then placed on the map depending on the longitude and latitude of the results for each LGA. The markers are binded to a popup which shows the name of the LGA along with the total number of the selected offence in the area. Refining data on this page was not implemented as I saw this page more about the location and less about the specific data.

There were not many problems during development of this page as I now understood the server request however I did have a problem with showing the markers on screen. My first method was to create a function that looped through the results and add markers to the map and call this function inside the return of the html. This didn’t work as no markers were placed. My second solution was to insert the marker placing code into the return of the map using {}. This worked and the markers were all placed with the popups working correctly. I then wanted to limit the markers to only show markers which had that crime reported in the area. This did not work and so the idea was scrapped.

Other Problems

One bug found is that when a user Searches data for a crime the loading page does not show sometimes. This can happen because the request to the server is satisfied quickly and the page does show the loading but so quickly its hard to see. The issue I see though is when the user clicks the search button the page does not change to loading but still takes some time to request the data and then will change to show the table. I was not able to figure out why this was occurring.

Testing and limitations

Throughout the project Manuel testing was used as each feature was developed to test the best case/working scenario. If the webapp worked as intended next task was developed. Once all tasks for the page were completed testing for unhandled exceptions began. For these test Functionality test were run to ensure the webapp handled these cases properly.

All test where run on Chrome, Firefox and Edge. Website was developed to be used at full screen on 100% zoom, This mainly affects styling.

Registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected output | Firefox | Chrome | Edge |
| Page Loading | Page loads Correctly with information shown | Pass | pass | pass |
| Correct Information Inputted | User is taken to login page | pass | pass | pass |
| Username already Used | User is told that registration has failed. | pass | pass | pass |
| Invalid character used | User is told invalid character is used. | pass | pass | pass |

Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected output | Firefox | Chrome | Edge |
| Page Loading | Page loads Correctly with information shown | Pass | pass | pass |
| Autofill after registration | After a user registers the forms for login are automatically filled | pass | pass | pass |
| Correct Information Inputted | User is taken to login page | pass | pass | pass |
| Invalid Username | User is told about the invalid username | pass | pass | pass |
| Invalid Password | User is told about the invalid password | Pass | Pass | pass |
| Invalid Character | User is told about an invalid character | fail | fail | fail |
| Invalid Character 2 (\*1) | User is told about an invalid character | pass | pass | pass |

(\*1) a change to the logic was made to show the correct message

Offences

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected output | Firefox | Chrome | Edge |
| Page Loading | Page loads Correctly with information shown | Pass | pass | pass |
| Request Fail | User is told something went wrong | pass | pass | pass |

Search

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected output | Firefox | Chrome | Edge |
| Page Loads | The page loads in from the tab selection correctly. | pass | pass | pass |
| Not logged in | The user is told they are not logged in | pass | pass | pass |
| Search button changes to loading screen | The user is shown that the page is loading | ½  Pass | pass | pass |
| The new page has all the correct information | The page has all the new indexes with a table | pass | pass | pass |
| Search for new crime | The table loads with new information. | pass | pass | pass |
| Refining shows correct refinement | The refine tags is updated and the table shows new information | pass | pass | pass |
| Logging out. | After the user logs in and back out the user is denied access | Pass | Pass | pass |

Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected output | Firefox | Chrome | Edge |
| Page loading | When page first loads it show correct information | Pass | Pass | pass |
| Not logged | The user is denied access if not logged in. | Pass | Pass | Pass |
| Fetching data | When the search button is hit the user is told that the page is loading | Pass | Pass | pass |
| Searched | The page shows the map with markers for LGA | Pass | Pass | Pass |
| Markers | Marker popup matches table data | Pass | Pass | Pass |
| New search | A new search updates the match data | Pass | Pass | pass |

References

Hand cuff logo: <https://dlpng.com/png/180192>

React guide: <https://devhints.io/react>

React Leaflet: <https://react-leaflet.js.org/docs/en/intro.html>

Leaflet: <https://leafletjs.com/>

Css: <https://www.w3schools.com/css/>

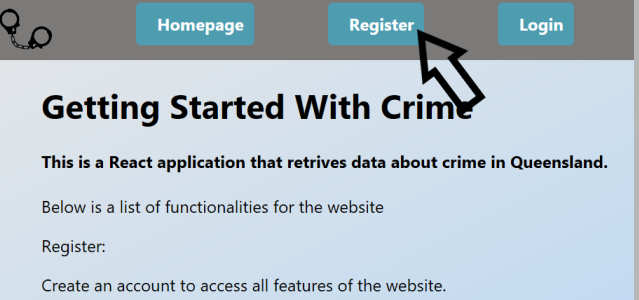
Routing: <https://auth0.com/blog/react-router-4-practical-tutorial/>

Appendices:

User Guide

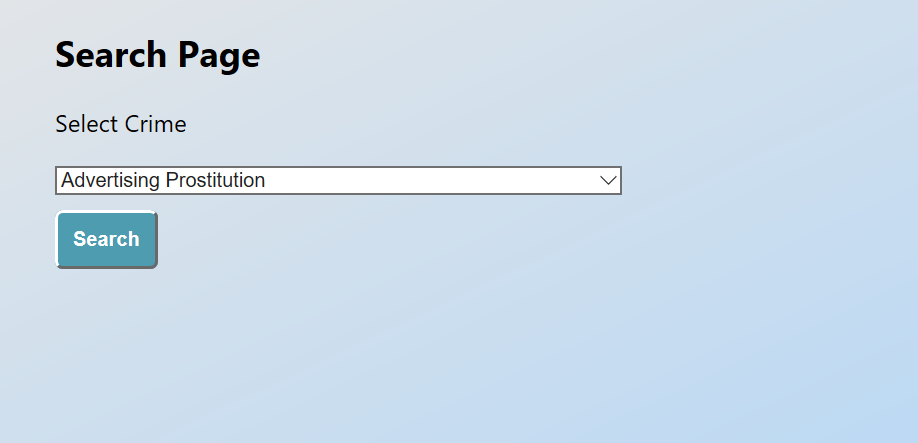
This is a small user guide to guide a new user through the Getting Started with Crime web application.

1. When the Web-app first opens you will be on the home page. This page gives a brief description of each part of the application.
2. Click on the Register Button at the top of the screen

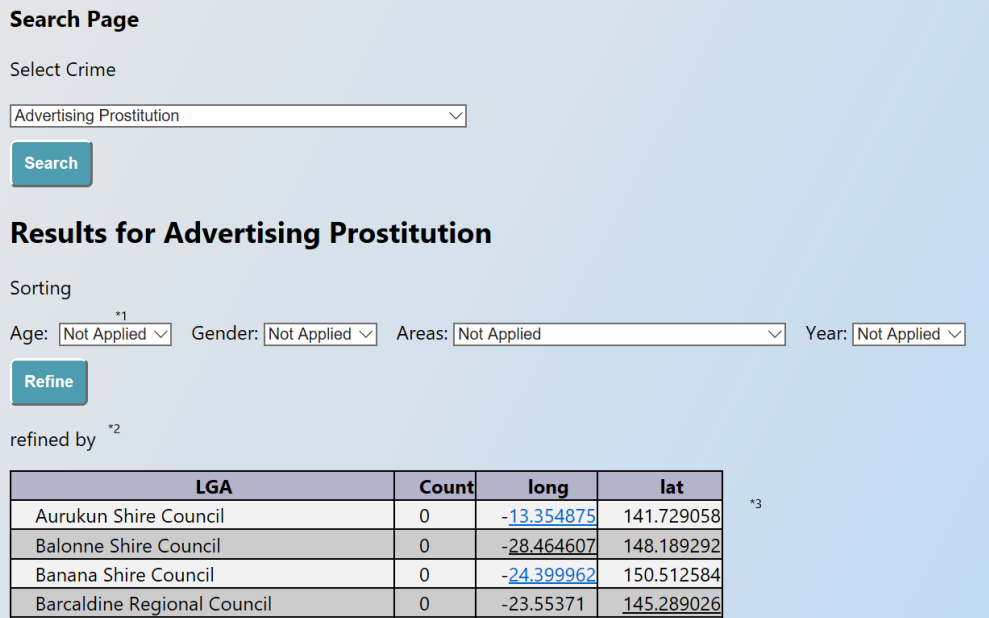


This will take you to the registration page

1. When on the registration page input a username and password in the directed fields and press register. If the username is taken it will show the Unsuccessful registration message, if this happens press the back button and try a new username and password. Do not use spaces in the username or password as this can break the application. You will be told if you have any invalid characters in the fields.
2. Once a new account has been registered you will be automatically taken to the login page and the details should be pre filled. Press the login button to continue.
3. Now that you have logged in you will be taken to the offences page. This page lists all of the currently registered offences in the system. You can look through this page but no extra functionality is there.
4. Click on Search button at the top of the screen to access a new page, if you see this page skip to step 8.



1. If you did not see the correct page you should have been told you were denied access, please try logging in again. the page requires users to be logged in to work so no information will be shown to users who are not logged in.
2. Once on the search page select the crime from the drop down menu you want to find information about. Once the crime is selected press the search button and wait for the page to reload. If something goes wrong you will be told something has gone wrong and given the option to refresh the application.
3. Now that the page has loaded you should see a page similar to below



1. With all the new information on the page explanations will be given about each new part.

\*1 The new drop down menus can be used to refine the data select the options you want to refine the data by then hit the refine button.

\*2 The refined by tag will update to show what current refinements are affecting the table.

\*3 the table shows the information retrieved about the crimes. LGA is local Government Area, Count is the number of total number of times the crime has been reported in the area. Long and lat are the longitude and latitude for the LGA

1. Select the new refinements you want to sort the data by the drop down menus. Press the Refine button to apply the new filters to the table.
2. Once done with the Search page Click on the map button at the top of the screen.
3. This will load a page similar to the search function page. Select which Crime you want to see from the drop down menu and click on the search button.
4. Wait for the page to load and now a map with markers will be shown on the page.
5. Each marker is placed on the long an latitude for the LGA, click on a marker to open a popup. This popup will show the name of the LGA along with the number of crimes reported in that area.
6. You can zoom in and explore the map to better separate the markers.
7. To log out press the logout button at the top of the screen.
8. Press the logout button that appeared in the page to log out of the account. You will know if you have logged out if you are taken to the login page and the form to login is available.

You have now explored every current feature in the application If any bugs occur a reload button should appear press the button to restart the service. If there is no button go to a different tab and back to retry.