CAB230 Client side report

Shannon Stephens, N10205144

# Introduction

For this assignment there was quite a split in terms of what was implemented and what was not implemented. This assignment implemented registration, logging in, a basic search functionality and a listing of all offences. The assignment also used React and url routing for the simple pages, though not for specific searches.

This assignment does not include any charts or maps, the search result table is a basic html table so you cannot sort by the column headers and registration details do not carry over to the login page.

There are also some aspects to the assignment that are either not entirely complete, don’t work properly or leave something to be desired. This includes: searches that include more than one area yield only the first result, you cannot search for more than one offence at a time, users need to manually type out search criteria correctly otherwise the search will not work, and there being no loading feedback for the search.

Much of the missing functionality was not implemented due to timing issues. I got very stuck very early in the process which hindered me from doing anything else. Once I got past that issue and got this part of the assignment to a passing stage I moved onto the server side. As this was rather late in the semester I did not get to implement any of the more advanced functionality or improve upon what was already there.

Below you can see logging in, viewing offences, viewing the search page, searching for an offences.

Figure 2 – viewing offences

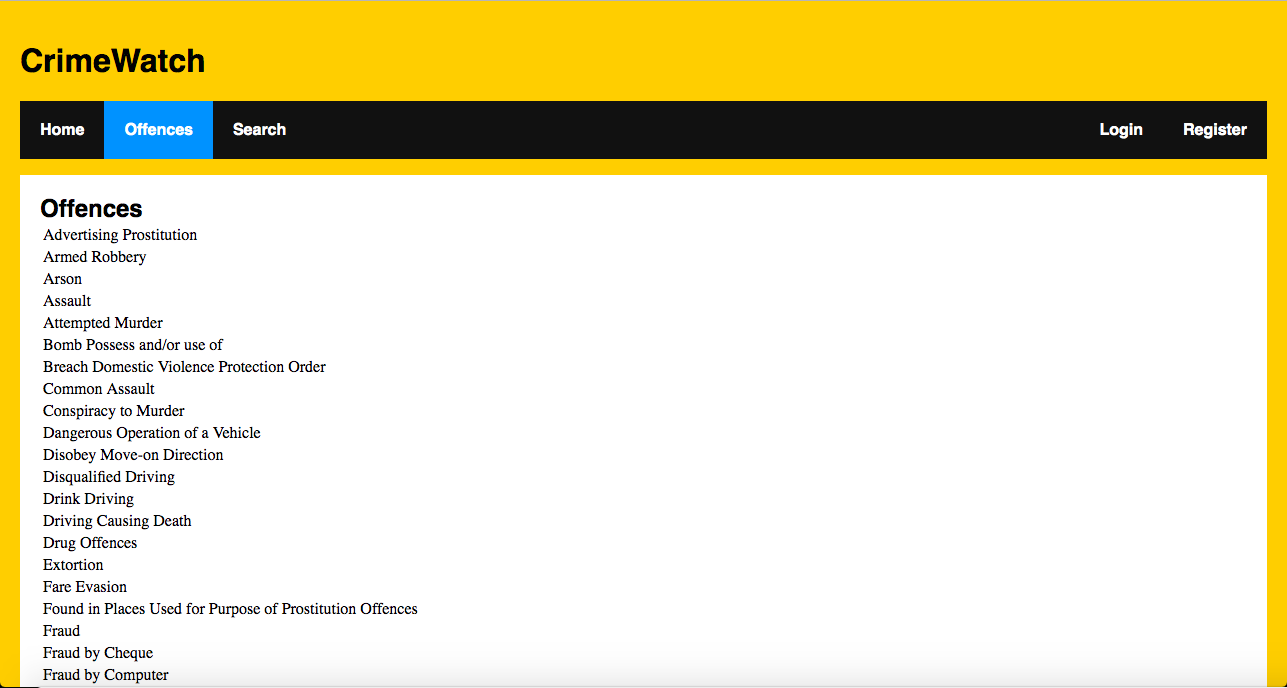
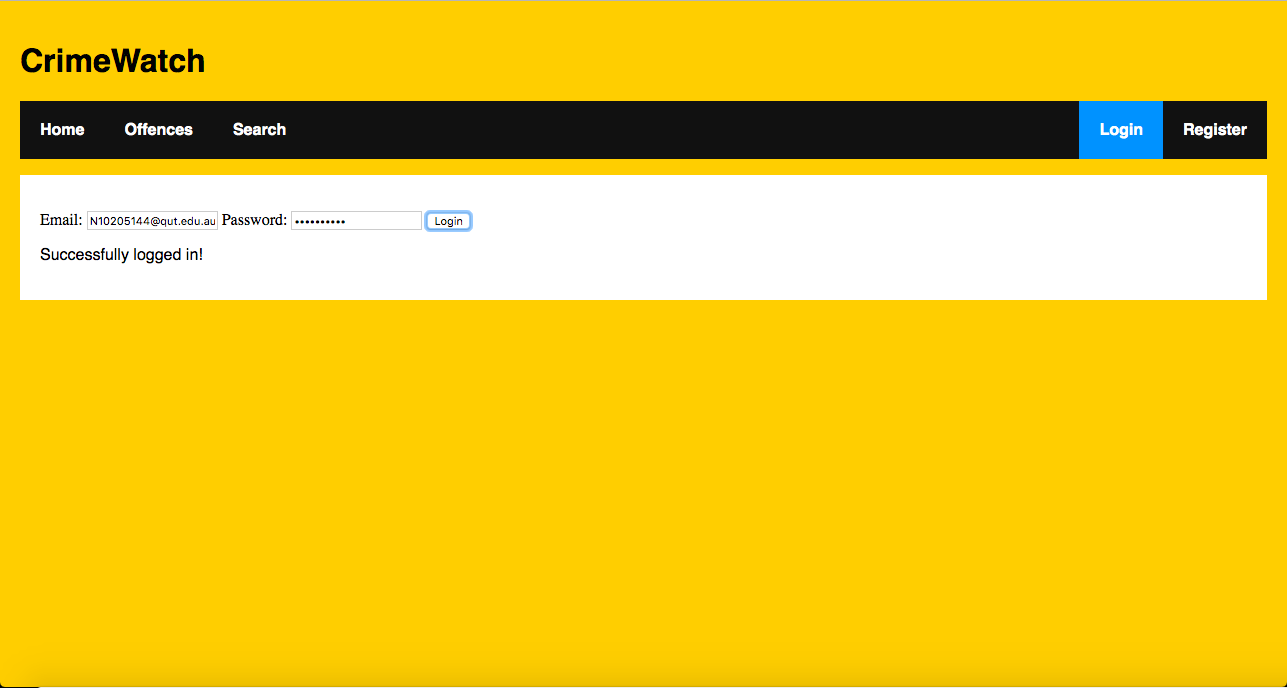


Figure 1 – logging in



Technical description of the application

Figure 4 – basic search

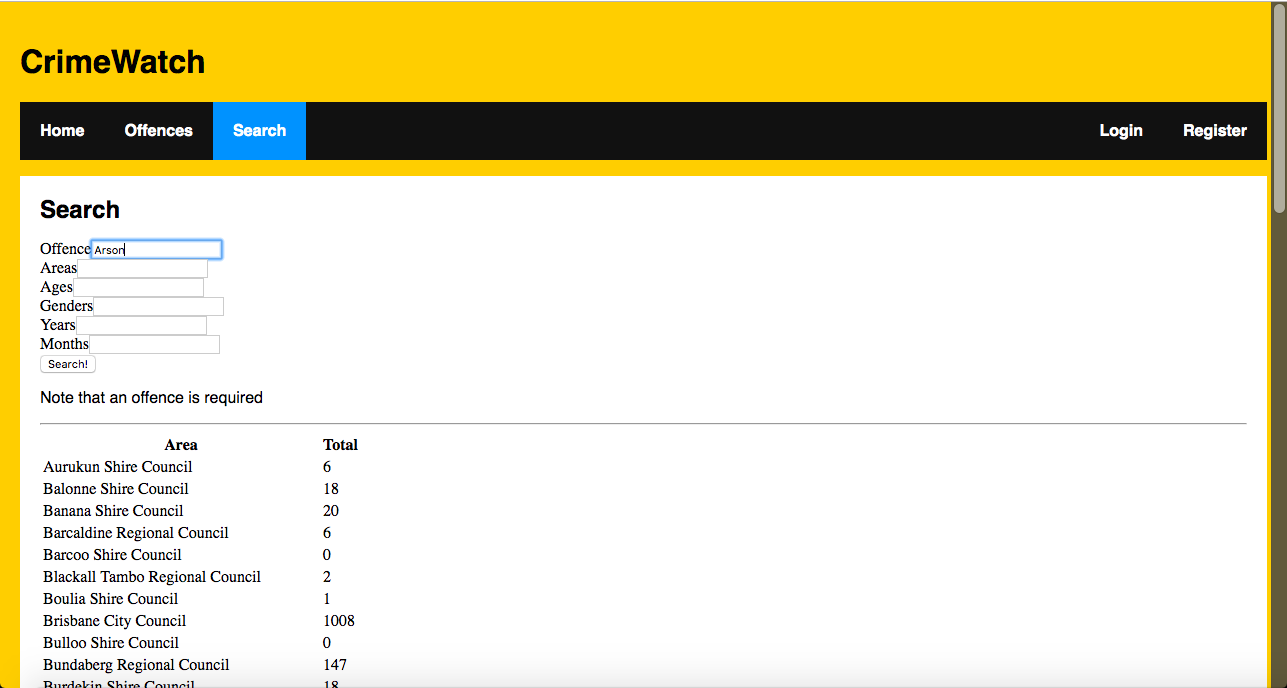
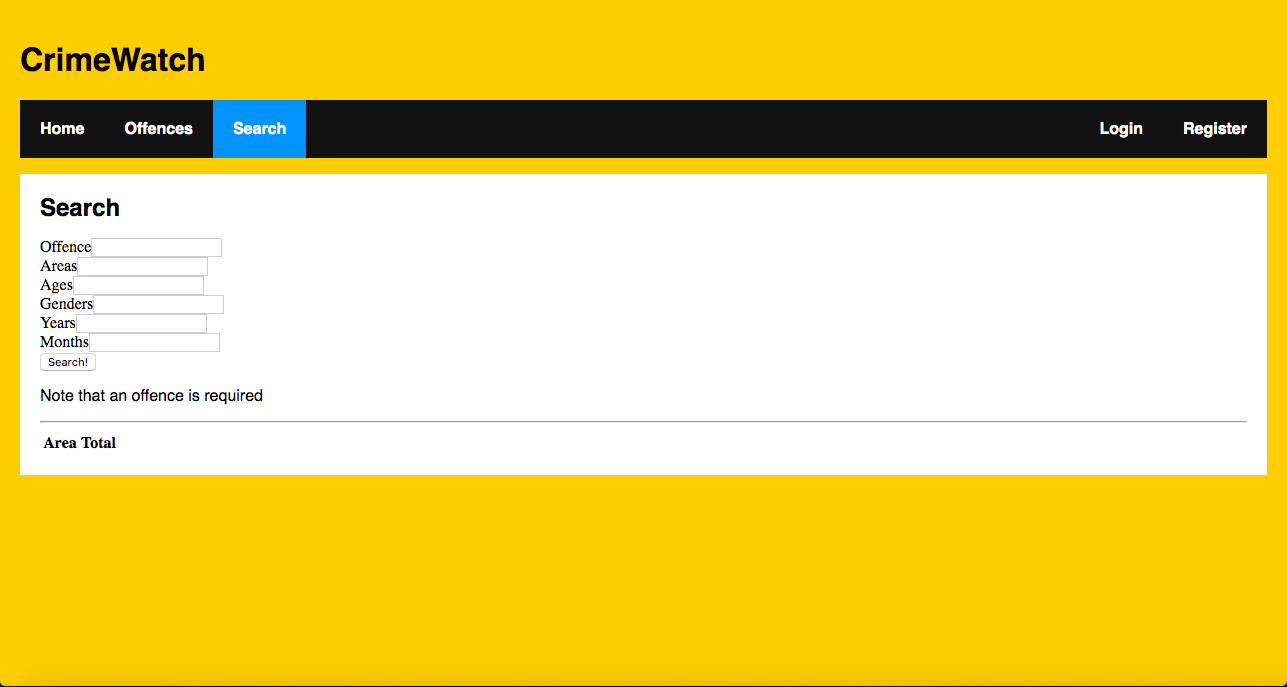


Figure 3 – viewing search page



Graphing and mapping was not implemented for this assignment.

The problem with the search queries involving area stumped me quite a bit because at first searching with area wouldn’t work at all. I found that in order to fix this I needed to replace commas and spaces manually instead of using a pre-made URI encoder. I did the same to area as I did to ages, genders, years and months and found that none of them actually seem to be displaying more than the first search criteria of their own field. So while the page doesn’t break the data displayed (and seemingly the data returned by the API) does not match up with the search criteria given by the user.

There was a similar problem with multiple offence searches, however instead of only displaying the first result there is no result given other than a 500 error. I could not find a work around for this as I discovered it quite late.

The big block I got stuck on as mentioned earlier in the report was mostly to do with general application structure and how different parts of the application interact, specifically react routers. I ended up finding a resource online which I based quite a portion of code off (Chinnathambi, n.d.). The demo they used that I based my code off had a particular look to it so I kept their CSS and page look as well.

# Testing and limitations

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality tested** | **Test setup** | **Expected result** | **Actual result** |
| Successful login | Login with a known registered account | “Successfully logged in!” displayed to the user and full functionality of implemented search | As expected |
| Bad login | Attempt to login with a garbage email and password | “Looks like there was a problem logging in” displayed to the user and cannot use search functionality | As expected |
| Successful registration | Register with an unused email and password that aren’t total garbage | “Successfully registered!” displayed to user and successful logging in with the same details | As expected |
| Bad registration | Register with a known used email and password | “Looks like there was a problem registering” displayed to user | As expected |
| Successful basic search | Search with only one offence parameter and no other search criteria | Results come back and are displayed in a table | As expected |
| Successful advanced search | Search with one offence but also some other search criteria | Results come back and are displayed in a table | For each search criteria (ages, genders, ect.) the search seems to only take the first argument into account. Data is displayed but seems to not match up entirely with the given search requirements. |
| Successful multiple offences search | Search with more than one offence | Results come back and are displayed in a table | 500 error |
| Bad search | Search with nonsense/garbage criteria | “Looks like there was a problem with your search” is displayed to the user | As expected |
| List offences | Open the Offences page | Offences are listed neatly | As expected |

# References

Chinnathambi, K. (n.d.). *Creating a Single-Page App in React using React Router.* Retrieved from Kirupa: https://www.kirupa.com/react/creating\_single\_page\_app\_react\_using\_react\_router.htm

# Appendix (user guide)

Welcome to CrimeWatch! Here you can view crime data in local government areas. First things first you will need to register if you haven’t made an account. To do this go to the Register page and enter the email and password you would like to use. After you have successfully registered you will also need to login.

Once you have an account go to the Login page and use the same email and password to login. Now you’re able to access all of the features in CrimeWatch.

To view a list of all offences go to the Offences page and there you should see a full list of all offences that CrimeWatch uses.

To look at some actual crime data head over to the Search page. Here all you need to do is type in an offence as it appears in the Offences page and hit search (please note that an offence is required). Now you should see a list of local government areas and the total number of times that offence was committed.

If you want your search to be more specific then type in some more search criteria into the other search fields and the list should update after you hit search again.