



---

# CAB230 CLIENT SPECIFICATION

---

Liam Wrigley – n9448381



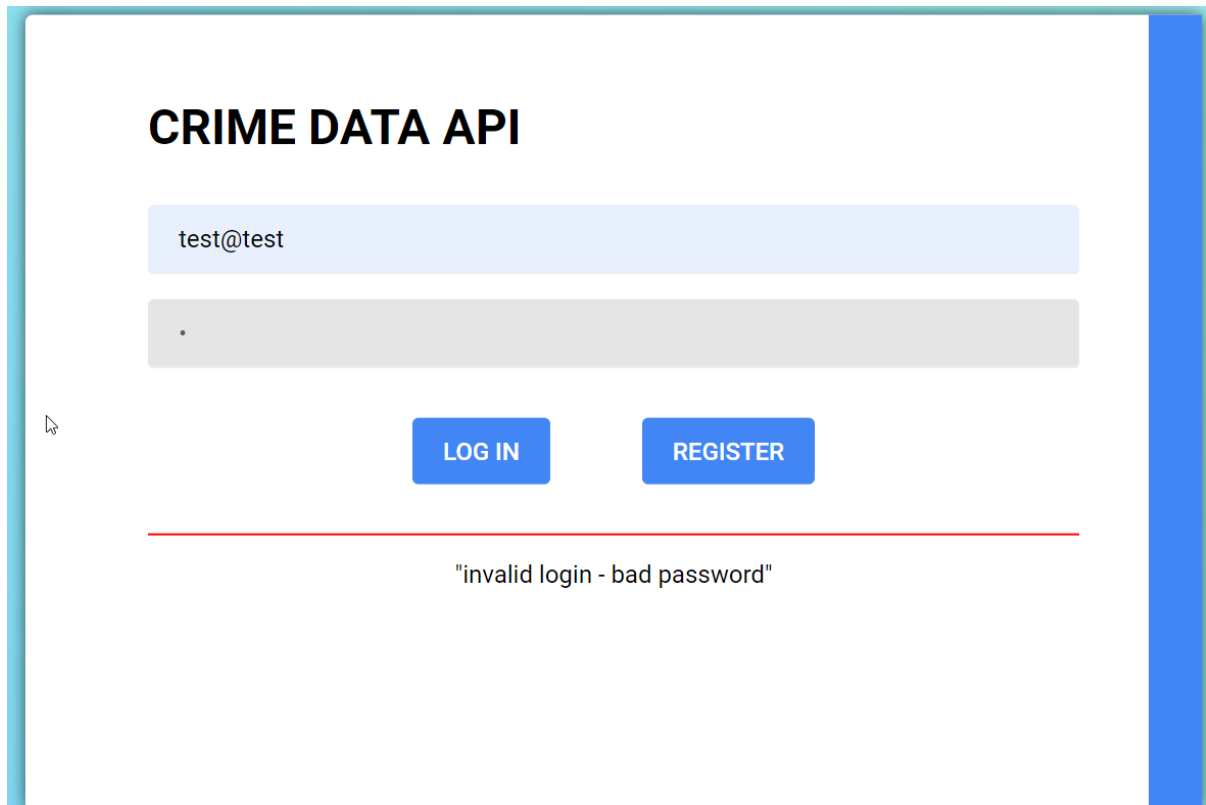
## 1. Introduction

Implemented:

- Made with react
- Ability to search all endpoints
- Login with successful error handling as well as saving auth token to local storage
- Logout button that will remove the auth token
- Register with proper error handling
  - Without auth token will be redirected to login page
- Table view with sortable headings
- Ability to sort by offence, area, age, gender, year and month
- Map view showing a heatmap of occurrences which is also sortable.
- Using HTTPS

Not Implemented:

- Graphing was not implemented
- Reset button for search not implemented



The screenshot displays a web interface for the "CRIME DATA API". It features a light blue header with the title "CRIME DATA API" in bold black text. Below the header, there are two input fields: the first is light blue and contains the email "test@test", and the second is light gray and contains a single dot, representing a password. Below these fields are two blue buttons labeled "LOG IN" and "REGISTER". A red horizontal line is positioned below the buttons, and underneath it, the error message "invalid login - bad password" is displayed in black text. The entire interface is framed by a light blue border, with a darker blue vertical bar on the right side.

Figure 1 Incorrect password

Lga ↕	Total ↕	Lat ↕	Lng ↕
Aurukun Shire Council	0	-13.354875	141.729058
Balonne Shire Council	0	-28.464607	148.189292
Banana Shire Council	0	-24.399962	150.512584
Barcaldine Regional Council	0	-23.55371	145.289026
Barcoo Shire Council	0	-25.031228	142.361338
Blackall Tambo Regional Council	0	-24.42259	145.463564
Boulia Shire Council	0	-22.753044	139.041312
Brisbane City Council	122	-27.470812	153.022455
Bulloo Shire Council	0	-27.996869	143.822286
Bundaberg Regional Council	3	-24.95816	152.262663
Burdekin Shire Council	0	-19.577286	147.406473
Burke Shire Council	0	-17.740903	139.545746
Cairns Regional Council	4	-17.01522	146.114225
Carpentaria Shire Council	0	-17.668838	141.080196
Cassowary Coast Regional Council	0	-17.973932	146.382423
Central Highlands Regional Council	0	-24.033049	148.737363
Charters Towers Regional Council	0	-20.07249	146.257893
Cherbourg Shire Council	0	-26.29018	151.95566
Cloncurry Shire Council	0	-21.249499	140.348184
Cook Shire Council	0	-14.281491	143.045893
Croydon Shire Council	0	-18.203489	142.243703
Diamantina Shire Council	0	-24.842624	139.691137
Doomadgee Shire Council	0	-17.63656	138.799512
Douglas Shire Council	0	-16.465232	145.37399
Etheridge Shire Council	0	-18.292413	143.543764
Flinders Shire Council	0	-21.01843	144.434214
Fraser Coast Regional Council	0	-25.488478	152.732388
Gladstone Regional Council	0	-23.842649	151.248859
Gold Coast City Council	43	-28.007483	153.344665
Goondiwindi Regional Council	0	-28.547261	150.30684
Gympie Regional Council	0	-26.183353	152.665746
Hinchinbrook Shire Council	0	-18.780386	147.101319
Hope Vale Shire Council	0	-15.294134	145.109552
Ipswich City Council	1	-27.695544	152.685365
Isaac Regional Council	0	-22.347632	147.281863
Kowanyama Shire Council	0	-15.472847	141.747512
Livingstone Shire Council	0	-22.437865	150.954007
Lockhart River Shire Council	0	-12.972868	143.131921
Locker Valley Regional Council	0	-27.627841	152.168857
Logan City Council	14	-27.774994	153.061857

Figure 2 Table view



Figure 3 Map view

## 2. Technical description

Figure 4 UseEffect to update any time a query part changes

The graphing was achieved using 'google-map-react' and uses the same data flow that the table view uses. The table is created using the package 'react-smart-data-table' and all data will update any time a dropdown value changes.

```

91   useEffect(() => {
92     getAllData(props).catch(e => {
93       setError(e);
94     });
95   }, [offenceQuery, areaQuery, ageQuery, genderQuery, yearQuery, monthQuery]);

```

The data is updated using a large string of variables that are updated any time a dropdown is changed.

```

116     <Dropdown type={"offences"} onQuerySelect={setOffenceQuery} />
117     <Dropdown type={"areas"} onQuerySelect={setAreaQuery} />
118     <Dropdown type={"ages"} onQuerySelect={setAgeQuery} />
119     <Dropdown type={"genders"} onQuerySelect={setGenderQuery} />
120     <Dropdown type={"years"} onQuerySelect={setYearQuery} />
121     <Dropdown type={"months"} onQuerySelect={setMonthQuery} />

```

Figure 5 Dropdown components in use

The dropdown component was created to dynamically create dropdowns of data that is fetched from within the component via the passed variable "type" seen in figure 5. The dropdowns each have an onchange event that will return current value encoded.

```

29     let items = dropdownData;
30     let optionItems = items.map(items => <option key={items}>{items}</option>);
31     return (
32         <select
33             onChange={event => {
34                 props.onQuerySelect(encodeURIComponent(event.target.value));
35             }}
36         >
37             {optionItems}
38         </select>
39     );

```

Figure 6 Creation of dropdown HTML

There is no graphing functionality for the data due to time constraints.

### 3. Testing and limitations – test plan, results, compromises.

#### Login:

INPUT	RESULT	EXPECTED
No email	"invalid login - you need to supply both an email and password"	"invalid login - you need to supply both an email and password"
No pass	"invalid login - you need to supply both an email and password"	"invalid login - you need to supply both an email and password"
Unregistered email	"oh no! It looks like that user doesn't exist..."	"oh no! It looks like that user doesn't exist..."
Wrong password	"invalid login - bad password"	"invalid login - bad password"
Successful	Redirect to data	Redirect to Data

#### Register:

INPUT	RESULT	EXPECTED
NO Email	It looks like that email is already registered	"error creating new user - you need to supply both an email and password"
No Password	It looks like that email is already registered	"error creating new user - you need to supply both an email and password"
Unique email and password	"yay! you've successfully registered your user account :)"	Successfully Registered!  Please log in

## 4. References

Packages used:

- react-smart-data-table – used for creating and showing tables
- google-map-react – used for mapping data on heatmap with Google API

## 5. Appendix: brief user guide

“npm install” to redownload all packages

“npm start” to start client.