CAB230 Web Computing Assignment Client Side Specification

Release Date: March 31 2019
Submission Date: May 31 2019 11:59PM
The submission is due the Friday of Week 13
Weighting: 60% of Unit Assessment
Task: Individual Project

Introduction:

In this assignment, your overall task will be to develop a web application that makes crime statistics available to the user, based on the data set that we have previously discussed in class and in the API guide. This initial document concerns the client side of the application, in which you will query the API and present the information in a form that is helpful to the user. Subsequently, you will be required to replicate the services that we are (and will be) providing on the server side, but it makes no sense to talk about that until we have at least introduced you to node.js.

The aims of this assignment are to:

- Introduce you to full web application development – both client and server side
- Provide experience in querying REST APIs and manipulating the results
- Provide experience in developing modern client applications
- Introduce you to developing modern data oriented REST APIs
- Provide experience in exposing a SQL database source as a REST API
- Provide experience in authentication and security
- Provide experience with application deployment
- Provide experience with modern web technologies including JS, React, Node.js and MySQL.

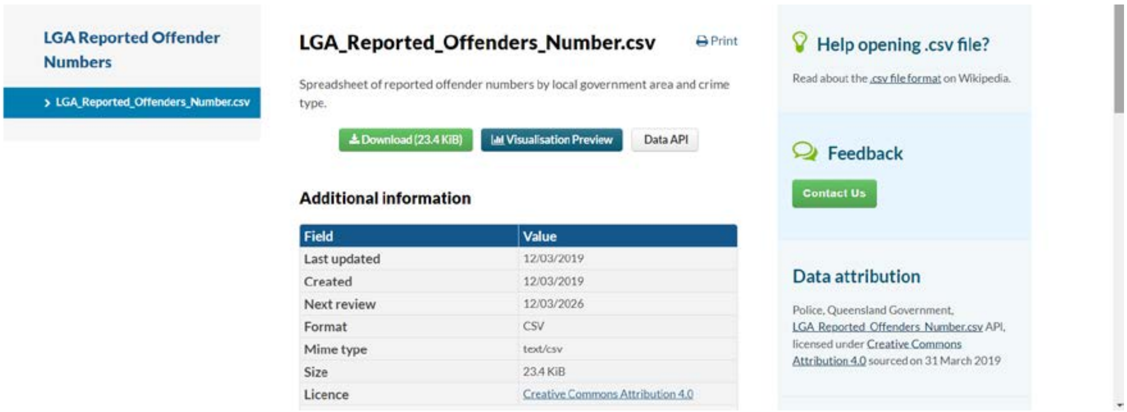
In this specification, we will not be too prescriptive, but we necessarily have to focus around some technologies, as otherwise we cannot support you in learning the material and debugging problems. You should see the development of the client for this application as a series of steps, with each one of those steps corresponding to an increase in sophistication, and in the possibility of a higher grade. We will consider these in turn below, and you can get more of an idea from the podcast which you find nearby on BB.

The Data:

The dataset is drawn from the Queensland Government open data initiative, showing criminal offences across the state over a period, organised according to offence type, local government area and type of offender. The data sources may be found at the following URL:

https://data.qld.gov.au/dataset/lga_reported_offender_numbers/resource/32d7bc11-55ed-4c52-86ff-e9d780cfe9ce

The page looks something like the image below.



LGA Reported Offender Numbers
› LGA_Reported_Offenders_Number.csv

LGA_Reported_Offenders_Number.csv

Spreadsheet of reported offender numbers by local government area and crime type.

Download (23.4 KiB)
Visualisation Preview
Data API

Additional information

Field	Value
Last updated	12/03/2019
Created	12/03/2019
Next review	12/03/2026
Format	CSV
Mime type	text/csv
Size	23.4 KiB
Licence	Creative Commons Attribution 4.0

Help opening .csv file?

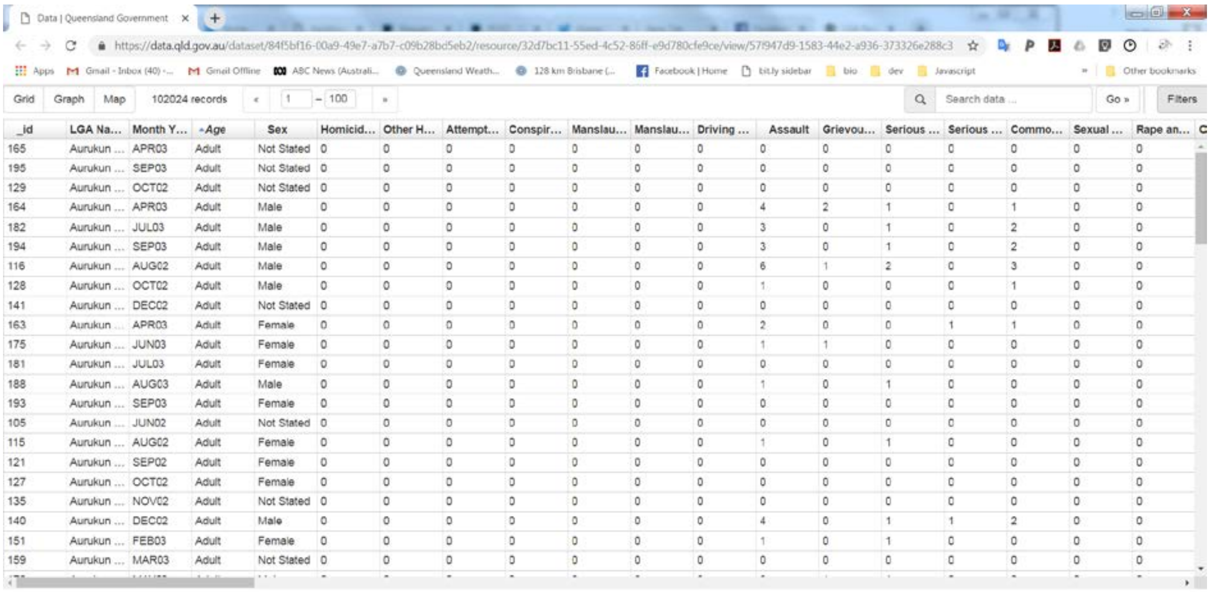
Read about the .csv file format on Wikipedia.

Feedback

Data attribution

Police, Queensland Government, LGA Reported Offenders Number.csv API, licensed under Creative Commons Attribution 4.0 sourced on 31 March 2019

When we preview the data – use the visualisation button – we will see that there are a massive number of columns.



_id	LGA Na...	Month Y...	Age	Sex	Homicid...	Other H...	Attempt...	Conspir...	Manslau...	Manslau...	Driving ...	Assault	Grievou...	Serious ...	Serious ...	Commo...	Sexual ...	Rape an...	C
165	Aurukun ...	APR03	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
195	Aurukun ...	SEP03	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
129	Aurukun ...	OCT02	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
164	Aurukun ...	APR03	Adult	Male	0	0	0	0	0	0	0	4	2	1	0	1	0	0	
182	Aurukun ...	JUL03	Adult	Male	0	0	0	0	0	0	0	3	0	1	0	2	0	0	
194	Aurukun ...	SEP03	Adult	Male	0	0	0	0	0	0	0	3	0	1	0	2	0	0	
116	Aurukun ...	AUG02	Adult	Male	0	0	0	0	0	0	0	6	1	2	0	3	0	0	
128	Aurukun ...	OCT02	Adult	Male	0	0	0	0	0	0	0	1	0	0	0	1	0	0	
141	Aurukun ...	DEC02	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
163	Aurukun ...	APR03	Adult	Female	0	0	0	0	0	0	0	2	0	0	1	1	0	0	
175	Aurukun ...	JUN03	Adult	Female	0	0	0	0	0	0	0	1	1	0	0	0	0	0	
181	Aurukun ...	JUL03	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
188	Aurukun ...	AUG03	Adult	Male	0	0	0	0	0	0	0	1	0	1	0	0	0	0	
193	Aurukun ...	SEP03	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
105	Aurukun ...	JUN02	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
115	Aurukun ...	AUG02	Adult	Female	0	0	0	0	0	0	0	1	0	1	0	0	0	0	
121	Aurukun ...	SEP02	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
127	Aurukun ...	OCT02	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
135	Aurukun ...	NOV02	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
140	Aurukun ...	DEC02	Adult	Male	0	0	0	0	0	0	0	4	0	1	1	2	0	0	
151	Aurukun ...	FEB03	Adult	Female	0	0	0	0	0	0	0	1	0	1	0	0	0	0	
159	Aurukun ...	MAR03	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

The REST API:

We have re-organised the data so that the endpoints can remain sane. The Data API button also allows the user to query via standard SQL syntax, but we have taken a slightly different approach, offering a search facility that does not require that the user know any SQL. As discussed in the API guide, we will offer the following endpoints:

- /register (POST) – Register using email and password
- /login (POST) – Logging into your account using the details provided. This allows access to authenticated routes via a JSON Web Token.
- /offences (GET) – returns an array of offences recorded. (Open endpoint).

- `/search?offence=xxx (GET) – The primary search route. Note that query params need to be url encoded. (Requires authentication).`

The `/search` route will shortly (early in the first week of April) support an optional `filter` parameter, with the obvious syntax:

- `/search?offence=xxx&filter=yyy (GET)`

Note that the `filter` parameter cannot be used by itself, and it must be coupled as shown with the offence parameter if the request is to be valid. The list of offences is available from the offences endpoint, and we will update the documentation on the main page at `http://hackhouse.sh:3000` to show these, and the valid values for the `filter` parameter. We will subsequently serve these endpoints via https, and the http addresses will disappear or be redirected. In the server side aspect of the assignment you will be required to implement the REST API and serve the content via HTTPS.

The Requirements:

The requirements for the client side of this assignment are ultimately pretty straightforward. The complications come because for most of you it is the first time that you have done something like this. At the most basic level, you must develop a client side web application that allows the user to work with each of the endpoints. The overall design of the site is up to you, but we will make some comments about clean and modern looking sites below.

Broadly speaking, the grade levels for this part of the assignment will correspond to the following feature levels. We will assume here of course that things are working perfectly, and you should understand that the grade levels can go down a little if there are errors or some elements are incomplete. But the basic passing levels are as follows:

4. A simple web page with limited formatting that implements the endpoint queries discussed below, and displays the results on the page using AJAX calls and HTML. As discussed in the podcast, much of this can be achieved by adapting the sample code from the API guide. Functionality must include proper registration, login and queries of the open and authenticated routes. React is not used at this level – although of course it is possible to get a 4 level mark by doing a bad job of a 5 level requirement. Similarly, if you do a bad job of this 4 level requirement, then you are likely to get a 3 level mark.
5. At the next level, we expect you to use React. The essential requirements here are that the responses to the queries should use the techniques that you will learn in the pracs of week 6 and week 7. We expect that the data tables should be readily sortable via the column headers, and this level includes successful processing of all of the endpoints discussed. For bonus credit you should include some in-page filtering – essentially you supply some text and only those results which match should remain.

We will give you some latitude in this, and you can also get some ideas from the discussion below.

6. At the higher grade levels, we will expect one or more features drawing on graphing and mapping. At the grade of 6 level, we will expect at least one of them to be implemented properly. We don't really care which, but the functionality should be in line with the examples discussed below. These components are visual in nature, and it is far better just to show you what we mean.
7. At the 7 level, you will need to implement both graphing and mapping, and to do both of them well. Ultimately we are not asking for more than a standard application of these libraries, but there is a fair bit of work for the novice to get on top of the data handling.

The endpoints are divided up in a couple of ways – how often they are used and whether or not they require authentication. We will consider each of them in turn.

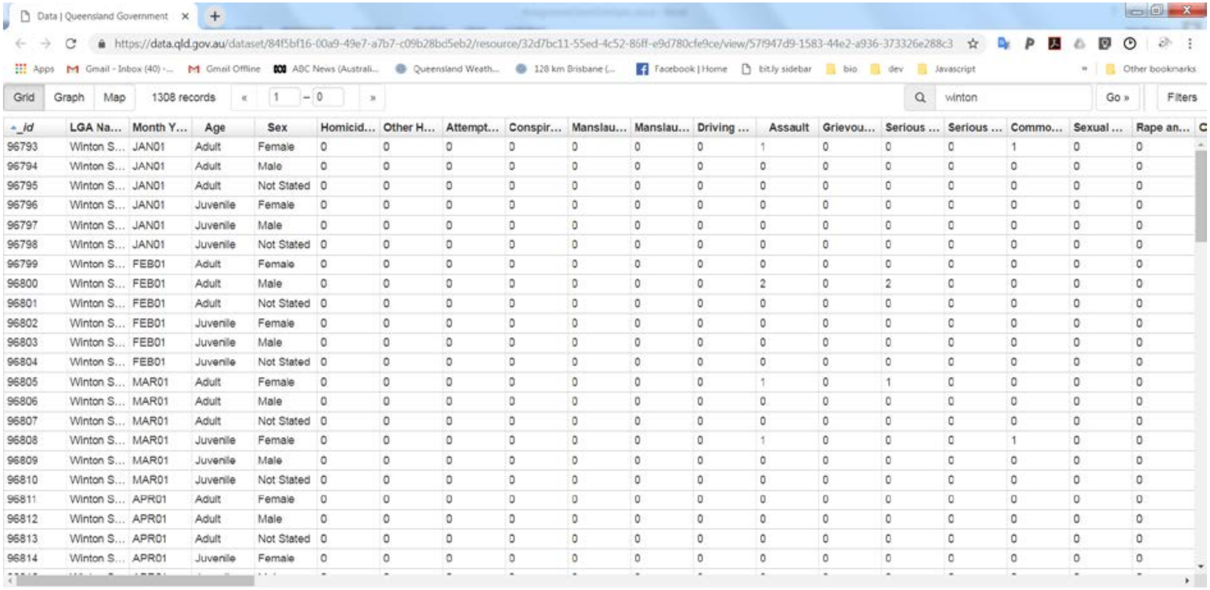
Registration and Login:
The `/register` and `/login` routes are best seen as prerequisites. We register only once, and we login once at the start of each usage period. You must use a form for each of these endpoints, as the user needs to supply an email and a password. The main elements of the form should be the same for each of them, and if the user has just registered, then their details should pre-populate the email and password fields for the login screen. Passwords should not be shown, and appropriate error messages should be displayed if the request fails. You should explore the response of the API to each of the obvious cases. If the email and password already exist in the database, then the registration will fail and the user needs to know this, and similar messages will be needed for errors involving login.

As in the example code from the API guide, you must propagate the login JWT in order to access the authenticated routes.

The Offences Information Endpoint:
The next endpoint is the simplest – an open `/offences` API that just lists the offence categories available in the database. Here, your basic task is simple – just display the offences in a table, with the functionality as listed above in the discussion of grade levels.

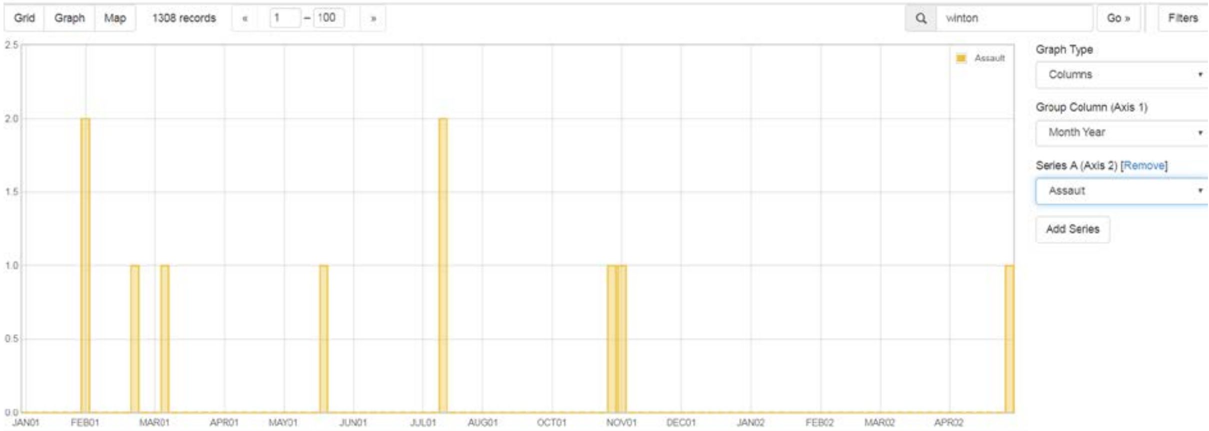
The Search Endpoint:
Finally, we get down to business. In this case, our basic level of functionality is to support the queries according to offence category with filters according to other attributes such as Local Government Area. These are extended as described above to reach the grade of 5 level. We now take a look at some possibilities for graphing and mapping and filtering. If we look at the grid display, we can quickly see a text box at the top, allowing us to search the data. On load, the data seems to be organised according to the ID field, and so at the top of the table I see mainly the entries from Aurukun and surrounding areas. If we type 'Winton' into the search

box at the top, then this has the effect of selecting specific records in the page. Note that this should not require that you hit the server again – this should be local processing of the collection of records that you have available. We see the result on the next page, dominated by Winton:



_id	LGA Na...	Month Y...	Age	Sex	Homicid...	Other H...	Attempt...	Conspir...	Manslau...	Manslau...	Driving ...	Assault	Grievou...	Serious ...	Serious ...	Commo...	Sexual ...	Rape an...	C
96793	Winton S...	JAN01	Adult	Female	0	0	0	0	0	0	0	1	0	0	0	1	0	0	
96794	Winton S...	JAN01	Adult	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96795	Winton S...	JAN01	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96796	Winton S...	JAN01	Juvenile	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96797	Winton S...	JAN01	Juvenile	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96798	Winton S...	JAN01	Juvenile	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96799	Winton S...	FEB01	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96800	Winton S...	FEB01	Adult	Male	0	0	0	0	0	0	0	2	0	2	0	0	0	0	
96801	Winton S...	FEB01	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96802	Winton S...	FEB01	Juvenile	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96803	Winton S...	FEB01	Juvenile	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96804	Winton S...	FEB01	Juvenile	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96805	Winton S...	MAR01	Adult	Female	0	0	0	0	0	0	0	1	0	1	0	0	0	0	
96806	Winton S...	MAR01	Adult	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96807	Winton S...	MAR01	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96808	Winton S...	MAR01	Juvenile	Female	0	0	0	0	0	0	0	1	0	0	0	1	0	0	
96809	Winton S...	MAR01	Juvenile	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96810	Winton S...	MAR01	Juvenile	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96811	Winton S...	APR01	Adult	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96812	Winton S...	APR01	Adult	Male	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96813	Winton S...	APR01	Adult	Not Stated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
96814	Winton S...	APR01	Juvenile	Female	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

We are now going to work with the graphs and the maps. Click on the graph button and make the selections as shown. Here axis 1 is the X axis, and axis 2 is the Y axis. We have selected assault as a column chart over time.



Graph Type: Columns

Group Column (Axis 1): Month Year

Series A (Axis 2) [Remove]: Assault

Here we see the assault figures for Winton over the period for which we have data. This is how your system should operate – we do a query and hit the server, we filter locally as well if we have that capability, and then we graph. You should support functionality at this level, but you may limit yourselves to column charts and line charts in which the X axis is time. You may use any of the standard JavaScript graphing libraries, and your code may be heavily based on their examples (hint – it always is). But you must tell us where it came from, and you still have the work required to make this talk to your data sources. Note that the line charts may not look all that effective for these data, and so we won't demand too much from you if you use them.

You may find the following links useful:

- Chart.js: https://www.chartjs.org/
- D3.js: https://d3js.org/

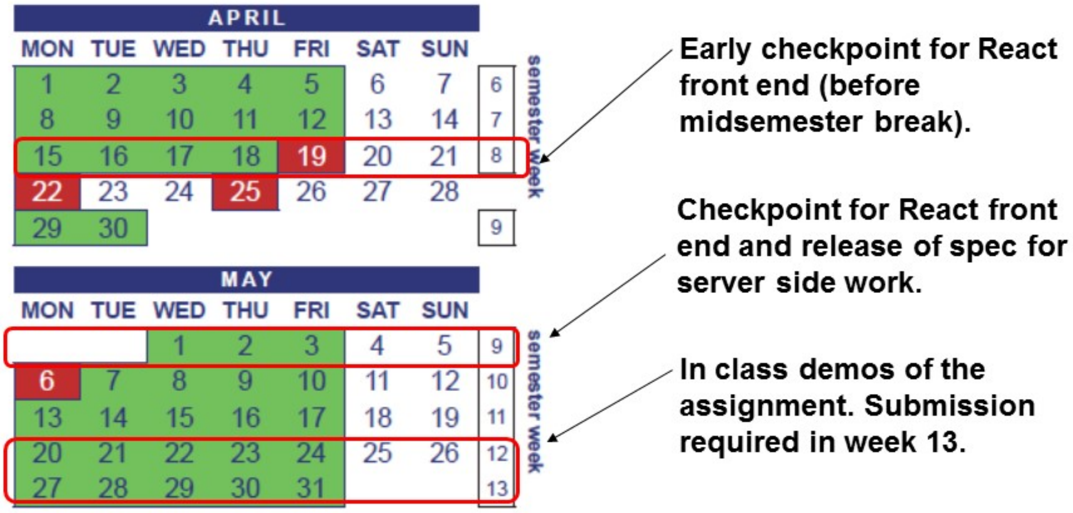
The approach to mapping is similar, but the Queensland Government page is not very intuitive. Here we will want you to show the relative levels for a particular time period for all the local government areas. Getting the full co-ordinates for a council region in Queensland is likely to be very tricky. We will update our SQL tables to include the latitude and longitude of the main centre for each Local Government Area, and requite that you place markers and/or a heat map blob on the map showing the values for the offences selected. We will give you an example once the tables are updated.

You may find the following links useful. We do not mandate a particular API, but many will use Google or Bing. ***Mapping products vary in their terms and conditions. It is not required that you purchase any product in order to complete this assignment. Some APIs may require your credit card prior to giving you an API key, but there are alternatives that don't require this. If you have any concerns, please talk to me at the lecture on Wednesday.***

- Google Maps: https://cloud.google.com/maps-platform/
- Leaflet JS library: https://leafletjs.com/
- Bing Maps: https://www.bingmapsportal.com/

Submission Requirements:

Which brings us finally to the submission. Your client application will be submitted as part of an overall package, which will include the server side code as well. Actual Blackboard submission will thus not take place until week 13. However, there are some steps that you will need to take in the interim. The image below is from the first lecture. The timetable remains unchanged in respect of the demos – you can demo to us before the holidays or just after, but you must demonstrate to us that your assignment actually works to a passing level. We will award 10 of the 60% allocated to the assignment if you can show us that you have a system that meets the requirements. If you do not demo, then the maximum you can get will be 50/60 overall. The purpose of this requirement is to encourage people to start early and to show us your work so that we can help you get on track for the highest mark possible.



APRIL

MON	TUE	WED	THU	FRI	SAT	SUN	semester week
1	2	3	4	5	6	7	6
8	9	10	11	12	13	14	7
15	16	17	18	19	20	21	8
22	23	24	25	26	27	28	
29	30						9

MAY

MON	TUE	WED	THU	FRI	SAT	SUN	semester week
		1	2	3	4	5	9
6	7	8	9	10	11	12	10
13	14	15	16	17	18	19	11
20	21	22	23	24	25	26	12
27	28	29	30	31			13

Early checkpoint for React front end (before midsemester break).

Checkpoint for React front end and release of spec for server side work.

In class demos of the assignment. Submission required in week 13.

If your demo is not satisfactory, your tutor will allow you to demo a second time up until the end of week 10. After that, the marks will be lost. The overall assessment of the assignment will be based on the server and the client side for 50 marks against the criteria sheets. There will be 30 marks attached to the server component and 20 attached to the client side. With 10 marks from the checkpoint demo, this will leave us with 60% overall.

There is one final requirement that you will need to look at before turning your attention to the server code. We will expect a short report and user guide, generally running to 5 pages or so, including screenshots. Your report must include the following sections:

1. Introduction – telling us what was implemented and what wasn't, showing a few screenshots to illustrate the functionality. This will probably occupy a page or a bit more.
2. Technical description of the application. This section is to allow you to talk about the APIs used for graphing or mapping, to show us any tricky data flows, and to discuss technical issues that caused you problems. This is especially important if something doesn't actually work.
3. Testing and limitations – test plan, results, compromises.
4. References
5. Appendix: brief user guide

We will require a similar report for the server side. For now, time to get working. The CRA rubric will be released as a separate file.