2023

<student name>

<student number>

1/1/2023

CAB230 Assignment 2 Client Side



Replace image with one with some relevance to your application here

CAB230

Movies API – Client Side Application

Contents

[Introduction 3](#_Toc131644938)

[Purpose & description 3](#_Toc131644939)

[Completeness and Limitations 3](#_Toc131644940)

[Use of End Points 4](#_Toc131644941)

[/movies/search 4](#_Toc131644942)

[/movies/data/{imdbID} 4](#_Toc131644943)

[/people/{id} 4](#_Toc131644944)

[/user/register 4](#_Toc131644945)

[/user/login 4](#_Toc131644946)

[/user/refresh 4](#_Toc131644947)

[/user/logout 4](#_Toc131644948)

[Modules Used 4](#_Toc131644949)

[Ag-grid-react 4](#_Toc131644950)

[Module 2 4](#_Toc131644951)

[Module n 4](#_Toc131644952)

[Application Design 5](#_Toc131644953)

[Navigation and Layout 5](#_Toc131644954)

[Usability and Quality of Design 5](#_Toc131644955)

[Accessibility 5](#_Toc131644956)

[Technical Description 6](#_Toc131644957)

[Architecture 6](#_Toc131644958)

[Test plan 7](#_Toc131644959)

[Difficulties / Exclusions / unresolved & persistent errors 7](#_Toc131644960)

[Extensions (Optional) 7](#_Toc131644961)

[User guide 7](#_Toc131644962)

[References 8](#_Toc131644963)

[Appendices as you require them 8](#_Toc131644964)

*This template is adapted from one created for a more elaborate application. The original author spends most of his professional life talking to clients and producing architecture and services reports. You may find this a bit more elaborate than you are used to, but it is there to help you get a better mark*

*This report should be around 10 pages or so including screenshots – there is no formal page limit, and the length will depend a lot on the number of screen shots, but you won’t get any extra marks for a really long report.*

## Introduction

### Purpose & description

This is written in a high-level professional tone. Tell is in about a paragraph or so what the app is supposed to do. This should be in ***your*** words, though you should feel *absolutely free* to steal some of it from the assignment specification.

In a second (and maybe third) paragraph, go ahead and tell us what to look for in your app: ***What did you do that was different? Did you do something to provide the user with functionality beyond what was expected? Is there some special set of modules that you have used that make it look great? Is there some other module that you have used that makes it more efficient?*** At this point, this description is at a very high level still. You will list your modules below.

At this point you can show 1-2 basic screenshots of your application to illustrate the approach, but leave the more detailed screenshotting to the use cases below.

### Completeness and Limitations

Here we want you to tell us in a couple of sentences what works and what doesn’t. ***Make a claim against the standards we laid out in the assignment specification (see below) and briefly justify that claim.*** Don’t give us deep details of the bugs here. Putting a positive spin on what you have achieved is fine – by all means focus on the stuff that works. But be realistic in your claim, as we will be looking closely at the app.

* **[Grade of 4 level]:** A simple React app with limited styling which uses the data endpoints and presents the data cleanly. Some basic navigation between pages is required. User endpoints and the authenticated data query may not have been implemented successfully. An attempt should be made at implementing a table component, but there may be limited use of pagination, and the user may be unable to filter the data. A react-strap table component or other simple alternative would suffice here.
* **[Grade of 5 level]:** At this level we would expect successful implementation of the user registration and login endpoints, and the authenticated data endpoint. Table components should utilise the standard functionality provided by a component such as AG Grid, and there should not be excessive querying of the server. The design of the website should be clean and generally uncluttered, although some parts of the website may be clumsy or confusing to use. An attempt may have been made at implementing a chart component, but shortcomings are likely to be present.
* **[Grade of 6]:** Here the expectation is that you have exceeded the grade of 5 level in that all the basics are there and working smoothly. Navigation is handled using React Router, controlled forms are used for inputs and there is evidence of close alignment between your chosen components and the data they are displaying. You are making use of refresh tokens so the user does not need to log in again every 10 minutes. The overall design of the website should be well thought out and it should be easy to use. We would also expect at this level for you to have used a chart component (or similar visualisation) on your ‘person’ page to show the IMDB scores of the films they worked on.
* **[Grade of 7]:** Our expectations for a grade of 7 are that you have covered everything needed for a grade of 6, and that you have gone beyond that, making full use in exposing the capabilities of the endpoints to users (e.g. with ag-grid infinite scrolling to handle the paginated search data.)

## Use of End Points

In this section we want you to show us the facilities that you have provided in the app. Here you should ***organize the discussion around the endpoints of the API, showing the screen corresponding to that endpoint and providing a brief discussion of what it doe***s. (A couple of sentences is fine here – the screen shot tells the story. Write more if there is something you want to tell us. But otherwise just keep it short.)

#### /movies/search

#### /movies/data/{imdbID}

#### /people/{id}

#### /user/register

#### /user/login

#### /user/refresh

#### /user/logout

## Modules Used

This is just a list of the external modules that you have used. You need not specify core React modules. In each case, ***we want the name, a brief description, and a link to the docs at npm or github or wherever.*** The example is ag-grid-react as most people will be using this. Just copy that style and add more as necessary.

#### Ag-grid-react

Module to provide fully-featured table components, including infinite scrolling.

<https://www.ag-grid.com/react-grid/>

#### Module 2

#### Module n

## Application Design

### Navigation and Layout

Here we want you tell us – again in a paragraph or two - about the design process for the site, the choices that you have made and any alternatives considered. Tell us about the choices you have made for navigation – the menu items and the flow between the screens – and the layout. The idea here is that you will tell us in this section how the application is used and we can then use this when we are looking at the technical description describing how it is built.

Show us some of your design mock-ups – scan some of your sketches – and show us how your application flows from one screen to another.

### Usability and Quality of Design

When we mark you on usability, ***our expectations are basic.*** Your application should be logically laid out and the UI widgets should be well chosen to suit the data that they control. This will ***not*** be a high bar to clear.

But in this section, we want you to be more critical and to assess – as firmly as you like – the quality of your own design work. We want you to highlight the good and the bad aspects of your design based on some of the principles you learned in the Web Design Principles Videos and Slides:

* Is your display well organized? Is the layout clean or clumsy?
* Is the navigation clear and intuitive?
* Is the application consistent with user expectations from other apps?
* Is the visual design consistent across screens? Do you use a small number of fonts and font sizes? Are there too many colours?

Comment on the usability of your design – are there compromises that make it awkward to use? How might you improve those?

Our expectation is that this section will occupy a few paragraphs or an equivalent number of bullet points. We are concerned more with the quality of the analysis than with the style.

### Accessibility

In this section we expect you to analyse your site from the perspective of accessibility, relying on our lecture slides and video, and the checklist from the W3C: <https://www.w3.org/TR/WAI-WEBCONTENT/full-checklist>. The documents from the W3C are dense and difficult to read, so we will work with an edited version of some of their most important, *Priority 1* requirements in the list below.

You do ***not*** need to implement accessibility features. Instead, we want you to examine your site critically. We want you to look at the checklist and, ***in a few paragraphs or an equivalent number of bullet points***, rate your app against these guidelines – you may even be able to tick off some of them – and tell us how your design will help or hinder you in meeting the requirements.

The edited list of Priority 1 Accessibility Requirements is here:

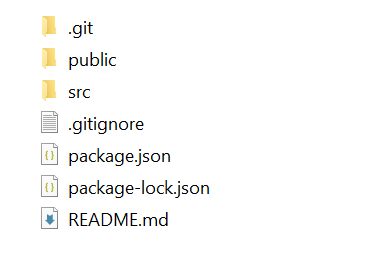
* Provide a text equivalent for every non-text element – alternatives to images, symbols, scripts, graphical buttons, sounds, audio and video files and so on.
* Ensure that all information conveyed with color is also available without color, for example from context or markup.
* Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.
* Ensure that text equivalents are updated when dynamic content changes.
* Avoid causing the screen to flicker.
* Use the clearest and simplest language appropriate for a site's content.
* For tables, identify row and column headers – clearly differentiated from the data.

You should feel free also to look at the lower priority guidelines available at the link above.

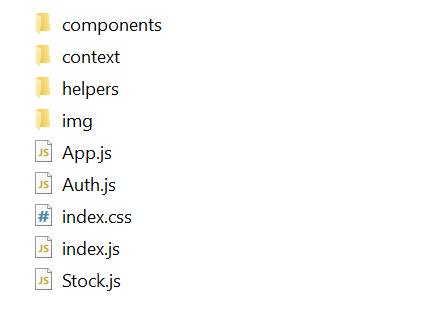
## Technical Description

### Architecture

Briefly describe the overall architecture of your application at a source code level. The description above tells us something of the application’s use. Now we want to see how that maps to the code organization. Your application source code will be organized something like this:



Dig into the src directory and tell us how you have split the responsibilities. Tell us in a sentence or two how the application is controlled and the services supported. In the image below we see an example organization of a React app. Include an image like this one and tell us briefly about the split across the relevant folders and why you chose it that way.



Overall, including the images, this section should still be under a page or so. We should get some sense of how the application works and how the data and control flows around. You may find it helpful to show us screen grabs of code if that makes your points clearer. Explaining quickly how the application works is the goal. Tell us anything you think we need to know about how you have structured the application and made it work, but there also a section below to describe problems.

### Test plan

Manual testing is fine and our expectations are in line with the example grid below. You can show the results through a screen shot and point us to these from the table.

Your tests should include

* Positive outcome cases
* Negative outcome cases (error scenarios)
* edge cases
* non-functional cases (ideally, but not required this time).

Note that the grid below is completely unrelated to this application.



Difficulties / Exclusions / unresolved & persistent errors /

Topics to include here could be:

* What were your major roadblocks / how did you resolve them?
* Any functionality you didn’t or couldn’t finish and the technical issues encountered
* Are there any outstanding bugs?

If you are describing bugs or unresolved issues please be quite specific and feel free to show screen grabs of code which will assist us in assessing your work.

## Extensions (Optional)

Where could you take this:

tell us about Potential future extensions / improvements for your app

## User guide

Tell us how to use your application

Use screenshots liberally here. You may re-use screenshots from above.

## References

Use a standard approach to referencing – see the guidance at <https://www.citewrite.qut.edu.au/cite/>.

## Appendices as you require them