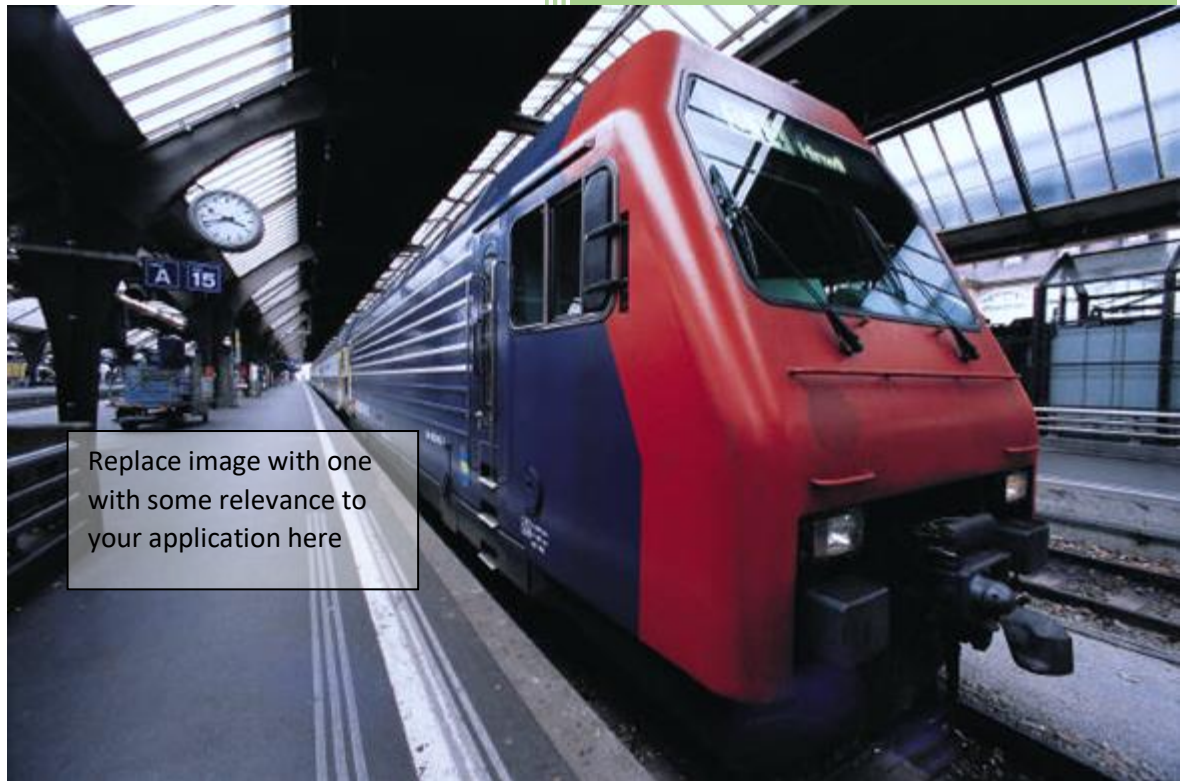


2020

CAB230 Stocks API – Client Side



Replace image with one
with some relevance to
your application here

CAB230

Stocks API – Client Side Application

<student name/s>

<student number/s>

4/23/2020

Contents

Introduction	2
Purpose & description	2
Completeness and Limitations	2
Use of End Points.....	5
/stocks/symbols.....	5
/stocks/{symbol}.....	5
/stocks/authed/{symbol}	6
/user/register	6
/user/login.....	6
Modules used	Error! Bookmark not defined.
Ag-grid-react.....	Error! Bookmark not defined.
Module 2	Error! Bookmark not defined.
Module n	Error! Bookmark not defined.
Application Design.....	8
Navigation and Layout	8
Technical Description.....	8
Architecture.....	8
Test plan.....	9
Difficulties / Exclusions / unresolved & persistent errors.....	9
Extensions (Optional).....	9
User guide	10
References	10
Appendices as you require them.....	10

Introduction

Purpose & description

Luma.io provides stock market data through an interactive and user-friendly experience. Its purpose is to provide users with an easy platform to view their chosen stock statistics. Due to it being a ReactJS based web application, Luma.io is a seamless and thought-out user experience for its consumers. While users enjoy this thought-out experience, it also allows users to easily view & query stock statistics from November 2019 all the way until March 2020.

Luma.io was designed to be user friendly and have a simple UI design. Although this is achieved somewhat with the provided CSS as far as styling, a lot of user functionality is lacking due to time constraints. These user functionalities would have exceeded the specification, but life events happen, and assignments have deadlines. This is unfortunate as my excitement to implement the power of ReactJS didn't get reflected in this assignment. Due to this, I will describe the little differences my application provides over the assignment specification.

One key difference implemented over the assignment specification clearly seen, is the Welcome User Message on the navigation bar. Alongside the logout button, it only displays on an authenticated navigation bar. Although this is equivalent to the use of the "authenticated navigation bar" in the client side example.mp4 by using a logout button.

luma.io

STOCK LIST

LOGINREGISTER

Stocks - LIST OF STOCKS

Show entries

10

Search

Symbol	Stock	Industry
A	Agilent Technologies Inc.	Health Care
AAL	American Airlines Group	Industrials
AAP	Advance Auto Parts	Consumer Discretionary
AAPL	Apple Inc.	Information Technology
ABBV	AbbVie Inc.	Health Care
ABC	AmerisourceBergen Corp	Health Care
ABT	Abbott Laboratories	Health Care
ACN	Accenture plc	Information Technology
ADBE	Adobe Systems Inc	Information Technology
ADI	Analog Devices Inc.	Information Technology

Showing 1 to 10 of 495 entries

Previous12345678Next

luma.io

STOCK LIST

Welcome userexample5LOGOUT

AMD - Advanced Micro Devices Inc

From:07/03/2020To:28/03/2020Filter

Timestamp	Symbol	Stock	Industry	Open	High	Low	Close	Volumes
24/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	40.62	42.32▲	38.95▼	41.64▲	100712000
23/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	41.51	42.685▲	39.58▼	39.61▼	106860000
20/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	39.56	42.2▲	37.69▼	39.62▲	88939000
19/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	39.54	41.98▲	36.75▼	39.12▼	106163000
18/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	40.19	42.08▲	38.3▼	41.64▲	92741900
17/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	39.08	42.02▲	38.51▼	38.71▼	84545900
16/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	42.2	43.01▲	39.6▼	42.0▲	86689700
13/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	42	42.5▲	39▼	39.01▼	107462000
12/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	44.19	45.97▲	43.8▼	45.7▲	102319000
11/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	45.41	45.41▲	43.1▼	45.38▼	76228200
10/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	43.03	46.34▲	42.51▼	45.27▲	80859800
09/03/2020	AMD	Advanced Micro Devices Inc.	Information Technology	49.44	49.21▲	46.24▼	48.53▼	124591000

luma.io

STOCK LIST

LOGIN

Not registered? Become a user today!

Enter your email:

userexample5

Enter your password:

Login User

2020 - cab230 assignment created by Dylan Buckley with <3

luma.io

STOCK LIST

REGISTER

Choose an email:

userexample5

Choose a password:

Register User

2020 - cab230 assignment created by Dylan Buckley with <3

Completeness and Limitations

I consider luma.io around 20% complete from the vision I originally had. Due to covid, I had a smaller amount of time to complete this assessment than anticipated. This result me in rushing the assignment and in turn only completing the basic functionality of the assignment specification. I was going to add a range of different charts & tables to view the API data the best way I could, but it took too long and had to be cut out due to time. Below is a quick list of some main components which are in it's current stages:

Completed:

- Complete authenticated query routes & unauthenticated query routes
- Client-side processing of table data
- React router to handle routing correctly
- Authentication detection for viewing stocks
- Registering Users
- Logging in Users
- Filter data between two query dates using JWT Authentication
- Display resulting date filter with a chart for closing price
- Display different table data based on authentication status

Partially Completed:

- CSS & Styling site wide. Some components such as the stock table and line graph contain CSS elements which are not completed with the current styling sheet. This is pagination & text color respectively.
- Login/Register Error Handling. Not error handling is displayed to the user, however HTTP status are logged in the console showing the status and result of each query. For example if a user is already registered, the console will display "User exists" instead of showing an error message to the user. These error conditions related to the login and registering of users aren't completed.
- Menu needs to show more information. Information including more navigational pages, user information, time information, and the current status of the API connected (as there is no way to tell if the API is connected besides not having table data)
- Table sorting by column needs to be labelled and have extended features. I would have ideally also added more sorting components for all columns, as well as row selection (currently handled by <a> tags around the stock symbol and name).

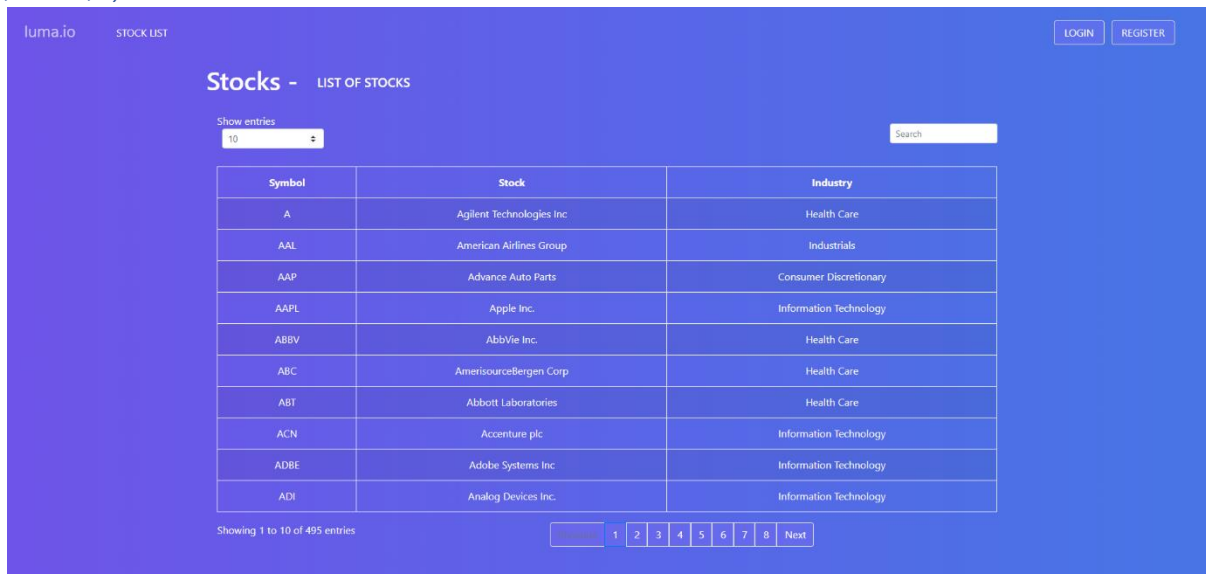
Not Implemented:

- Open, High, Low and Volumes line charts could be added.
- Optimize algorithms (No conditional rendering)
- Error handling executed on the user interface. Such as errors with users or fetching data.
- React Forms
- Try & Catch error handling for requests & conditions
- No Re-Renders (a.k.a the use of windows.location.href to update table when filtering)
- Beautiful code :'(

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 does**. (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.)

</stocks/symbols>



The screenshot shows the 'Stocks - LIST OF STOCKS' page. It features a search bar and a table with 10 entries. The table has three columns: Symbol, Stock, and Industry. The data is as follows:

Symbol	Stock	Industry
A	Agilent Technologies Inc.	Health Care
AAL	American Airlines Group	Industrials
AAP	Advance Auto Parts	Consumer Discretionary
AAPL	Apple Inc.	Information Technology
ABBV	AbbVie Inc.	Health Care
ABC	AmerisourceBergen Corp.	Health Care
ABT	Abbott Laboratories	Health Care
ACN	Accenture plc.	Information Technology
ADBE	Adobe Systems Inc.	Information Technology
ADI	Analog Devices Inc.	Information Technology

Below the table, it says 'Showing 1 to 10 of 495 entries' and provides pagination controls (Previous, 1, 2, 3, 4, 5, 6, 7, 8, Next).

This API endpoint allows unauthenticated users to view the stocks in a table. Therefore this data is displayed on the homepage for any user to see, including authenticated ones.

</stocks/{symbol}>



The screenshot shows the 'AAPL - Apple Inc.' page. It features a red message: 'Login to view this authorized data'. Below this is a table with three columns: Symbol, Stock, and Industry. The data is as follows:

Symbol	Stock	Industry
AAPL	Apple Inc.	Information Technology

At the bottom of the page, there is a footer: '2020 - cab230 assignment created by Dylan Buckley with <3'.

This API endpoint allows unauthenticated users to view a certain stock from the API. They can access this by clicking a stock name or symbol on the user home page.

/stocks/authed/{symbol}

The screenshot shows the 'STOCK LIST' page on luma.io. The header includes the logo 'luma.io', the text 'STOCK LIST', a welcome message 'Welcome userexample5', and a 'LOGOUT' button. The main content area is titled 'ADBE - Adobe Systems Inc'. Below the title, there are date range filters: 'From: 07/03/2020' and 'To: 28/03/2020', with a 'filter' button. A table displays stock data for ADBE on 24/03/2020. The footer contains the text '2020 - cab230 assignment created by Dylan Buckley with <3'.

Timestamp	Symbol	Stock	Industry	Open	High	Low	Close	Volumes
24/03/2020	ADBE	Adobe Systems Inc	Information Technology	296.67	310.95 ▲	287.78 ▼	307.51 ▲	5865950

This API endpoint allows authenticated users to view a certain stock from the API. The provides the certain stock requested along with extra data only accessible through a JWT authenticated request.

/user/register

The screenshot shows the 'REGISTER' page on luma.io. The header includes the logo 'luma.io', the text 'STOCK LIST', and buttons for 'LOGIN' and 'REGISTER'. The main content area is titled 'REGISTER'. It features two input fields: 'Choose an email:' with the value 'userexample5' and 'Choose a password:' with masked characters '***'. A 'Register User' button is located below the password field. The footer contains the text '2020 - cab230 assignment created by Dylan Buckley with <3'.

This API endpoint allows registering for unauthenticated users.

/user/login

The screenshot shows the 'LOGIN' page on luma.io. The header includes the logo 'luma.io', the text 'STOCK LIST', and buttons for 'LOGIN' and 'REGISTER'. The main content area is titled 'LOGIN' with a sub-header 'Not registered? Become a user today!'. It features two input fields: 'Enter your email:' with the value 'userexample5' and 'Enter your password:' with masked characters '***'. A 'Login User' button is located below the password field. The footer contains the text '2020 - cab230 assignment created by Dylan Buckley with <3'.

This API endpoint allows unauthenticated users to log in.

Modules

- **MDBReact**

MDBReact was utilized for the stock statistics tables. I personally found MDBReact over agGrid to be simpler to implement along with customization of the CSS. Although I haven't completed the all of the CSS for my MDBReact table, as you can see by the pagination, it's flexibility and simplicity helps with all around sorting and filtering. I can show the number of entries, with pagination and a column search with its Table single table function.

<https://www.npmjs.com/package/mdbreact>

- **React-chartjs-2**

For viewing the closing price line graph, the line graph component from the chart.js library was used. In combination with a MDBReact container to position the graph, the graph shows the closing price just like the assignment specification. The CSS styling has not been completed for this component and therefore the color of the text is hard to see. Like all uncompleted CSS styling, this chart was hard to reference in my stylesheet, so it didn't allow for a quick style within the time limit.

<https://www.npmjs.com/package/react-chartjs-2>

- **React-Router**

React-router was used to handle the navigation of the Luma.io application. I originally had <ProtectedRoute> components which allowed pages to be authenticated. This means that a user could not visit the /stocks/ page without authentication. After reading the assignment spec I found that following the guide of allowing users to go to /stocks/ but have a limited view was the better option.

I also implemented clean URL routes with parameters. This goes outside the assignment specification due to my preference. This feature allows URL's to be display cleanly. For example, a URL parameter which contains the ABT stock would usually look like: /stocks?=ABT, however I have cleaned that up to look like: /stocks/ABT.

<https://www.npmjs.com/package/react-router>

- **ReactStrap**

Reactstrap was used as the CSS framework used to style all the CSS found on luma.io It is used for the button hovering animations, along with many predefined styles found in the Bootstrap 4 framework. However due to time constraints as mentioned, a lot of the CSS was done by hand and hardcoded. It is not built as responsive web application for mobile devices.

<https://www.npmjs.com/package/reactstrap>

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.

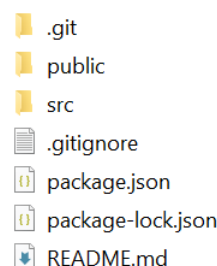
Show us some of your design mock-ups – scan some of your sketches – and show us how your application flows from one screen to another. ***Comment on the usability of your design – are there compromises that make it awkward to use? How might you improve those? Note that our expectations in respect of usability are basic – you can be critical here without losing lots of marks.*** We are looking for you to highlight the good and the bad aspects of your design and layout.

Please see the CRA and podcast 2 for more guidance, but your application should be logically laid out and the widgets should be well chosen to suit the data that they control. This will **not** be a high bar to clear.

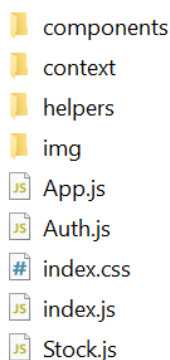
Technical Description

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 the organization for Michael's demo app. Tell us briefly about the split across these folders and why you chose it that way.



Overall, including the images, this section so far should still be under a page. We should get some sense of how the application works and how the data and control flows around. You may also find it helpful to show us screen grabs of code if that makes your points clearer. 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 unrelated to this application.

Task	Expected Outcome	Result	Screenshot/s (Appendix B)
Search for images	Results displayed, Map populated	PASS	01
Sort search results	Results sorted as per selection	PASS	02
Limit number of results	Result count limited as per selection	PASS	03
Click down arrow to load next page	Next set of results displayed	PASS	04
Click result to view info	Popup displayed with photo info	PASS	05
Click marker to view info	Popup displayed with photo info	PASS	06
Click popup image to view on Flickr	Flickr image opened in tab	PASS	07
Click user icon to view Flickr profile	Flickr user profile opened in tab	PASS	08
Click location to search	Search executed using location	PASS	09
Click tag to search	Search executed using tag	PASS	10
Click Google Vision location to search	Search executed using location	PASS	11
Click Google Vision tag to search	Search executed using tag	PASS	12
Handle Flickr response/format error	Display popup, Application continues	PASS	13
Handle GVision response/format error	Display 'Error', Application continues	PASS	14

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

References

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