

2020

CAB230 Stocks API – Client Side



Declan Barrett

N10219358

5/10/2020

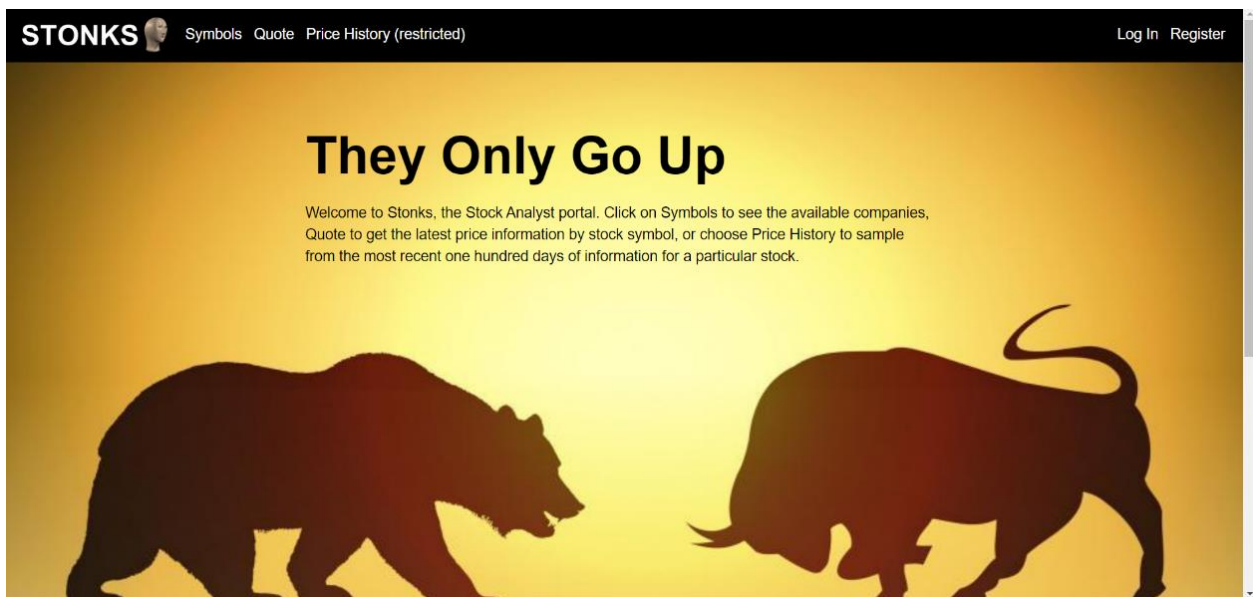
Contents

Introduction	2
Purpose & description.....	2
Completeness and Limitations.....	3
Use of End Points	4
/stocks/symbols	4
/stocks/{symbol}	4
/stocks/authed/{symbol}	4
/user/register	5
/user/login	5
Modules used.....	5
Ag-grid-react	5
Bootstrap	5
Reactstrap	5
Chartjs	5
React-Chartjs-2.....	6
Chartjs-plugin-zoom.....	6
jsonwebtoken.....	6
Application Design	7
Navigation and Layout	7
Technical Description.....	8
Architecture	8
Test plan.....	8
Difficulties / Exclusions / unresolved & persistent errors.....	9
Extensions	9
User guide	9
References	14
Appendix	15

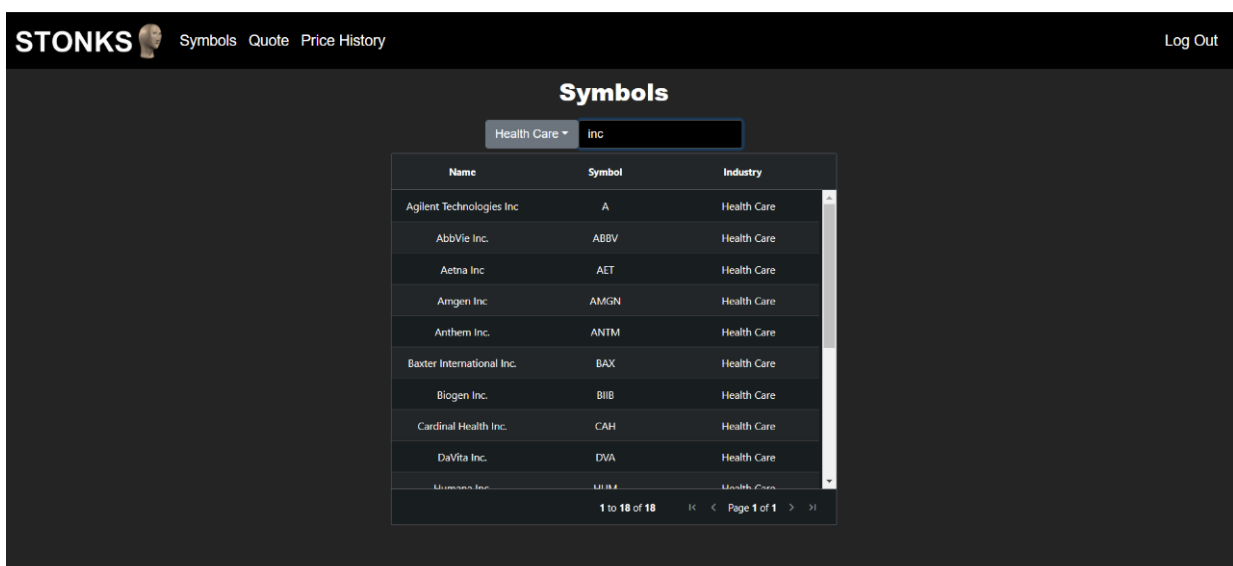
Introduction

Purpose & description

The STONKS stock analyst app is designed to allow users to view and analyze stock market statistics drawn from a database which is exposed via a REST API. Through all this the user is meant to get a clean, easy to use, experience enhancing their ability to buy and sell stocks with additional information. Users can navigate between multiple different dynamic pages in search of information pertaining to the stock market. This includes a search by industry and stock name to find specific symbols that users are looking for and a “Quote” which gives them the latest prices and volume of the stock. The price history of a stock is also available once the user has logged in. Once given a range of dates and the symbol, the price history produces all price and volume information on the stock between those dates. The user can log in, log out and register via their email and password.



The site is filled with functionality, including allowing users to view the price and volume of stocks in a graph format on the same graph in the price history tab. This chart also has the ability to zoom in and out, pan, add or remove sets of data that the user doesn't want to include. The price history tab is also able to be directly accessed from the symbols tab via clicking on the symbol desired. If the user

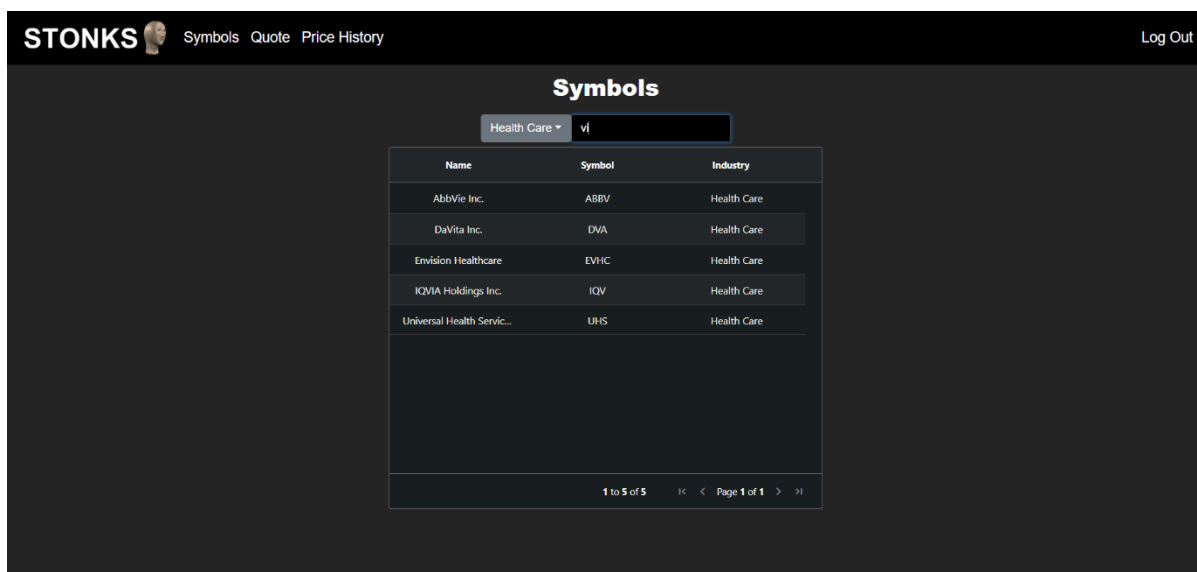


has logged in, the symbols tab will display the price history and all its information, which can then be adjusted. If the user isn't logged in then they are redirected to the log in screen and told that it is a user only feature and they need to sign in to access the content. The forms of the app are sleek in design and give user feedback for incorrect inputs and the header buttons adapt to whether the user is logged in or not. The header also formats differently depending whether its in full-size or small window mode.

Completeness and Limitations

The STONKS app is a React app with themed styling which cleanly presents its data in ag-grid-react table components. This data comes from both the authenticated and unauthenticated query routes with all end-points having been implemented. Users can register for an account which logs them into the account, or separately log into that account, with authentication being assessed and access granted to price history based upon it. Client-side processing via both filter functions and ag-grid-react are used, including sorting and filtering, as well as server-side filtering via additional query parameters. No excess querying of the server occurs, with requests only sent when new information is placed in forms and the forms contain information to send.

Navigation around the site is handled through React Router, with Route, redirects, history and link all used in appropriate cases. Reactstrap forms and input groups are used for data entry and every input is positioned appropriately and is of the type of input which is best suited for the data being inputted, such as date gathers to gather dates. Chartjs with its react wrapper are used to display the price and volume data in the price history tab, also adapting to the users choice of dates and symbol. Additionally, the header of the app dynamically adapts to the users state of log in, the symbols table links to the price history tab and displays the symbol that was selected and the chart in price history can be zoomed in, panned and have the user customize which data set they would like to see. All errors from the server and client are handled and in an appropriate format and are told to the user so they can have an easier time interacting with forms. The site itself is themed to the stock market, with the front page, header title, logo and name of the site being in reference to the widely popular stock trading meme. Overall, based on the assignment's specification and the current state of the STONKS web application, the web application completely fulfills the requirements of a grade of 7.

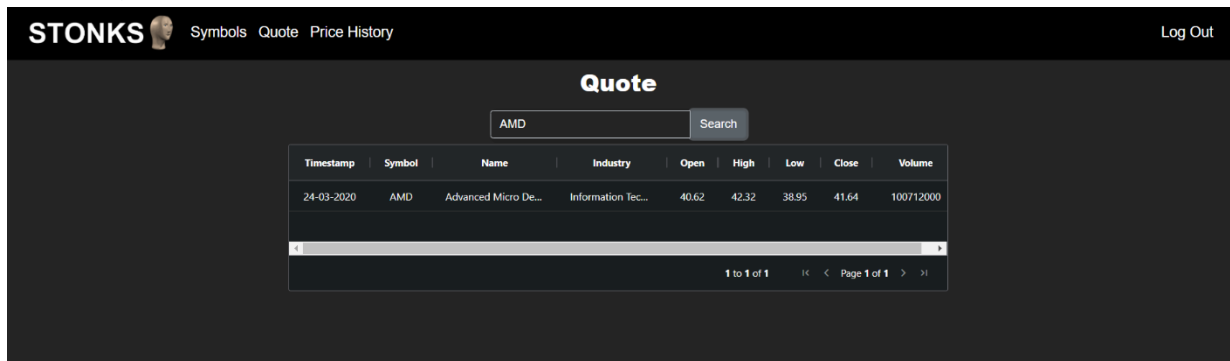


Use of End Points

</stocks/symbols>

The symbols endpoint is used for the Symbols tab. By default, it shows the 'all' request which displays all of the industries and their symbols. Changing the industry via drop down queries the symbols end point for the specific industry. The stock name input can be used to filter client-side within the currently selected industry for the desired company. The data is placed in a table and is able to be sorted, and clicking a symbol links to the price history page.

</stocks/{symbol}>



The screenshot shows the 'Quote' page for AMD. The page has a dark theme with a navigation bar at the top containing 'STONKS', 'Symbols', 'Quote', and 'Price History', along with a 'Log Out' link. A search bar contains 'AMD' and a 'Search' button. Below the search bar is a table with the following data:

Timestamp	Symbol	Name	Industry	Open	High	Low	Close	Volume
24-03-2020	AMD	Advanced Micro De...	Information Tec...	40.62	42.32	38.95	41.64	100712000

At the bottom of the table, it indicates '1 to 1 of 1' and 'Page 1 of 1'.

The symbols endpoint is used for the quote tab. By default, it only shows a form with the word 'symbol' and an empty table. Putting in a symbol and submitting produces information in the table

</stocks/authed/{symbol}>



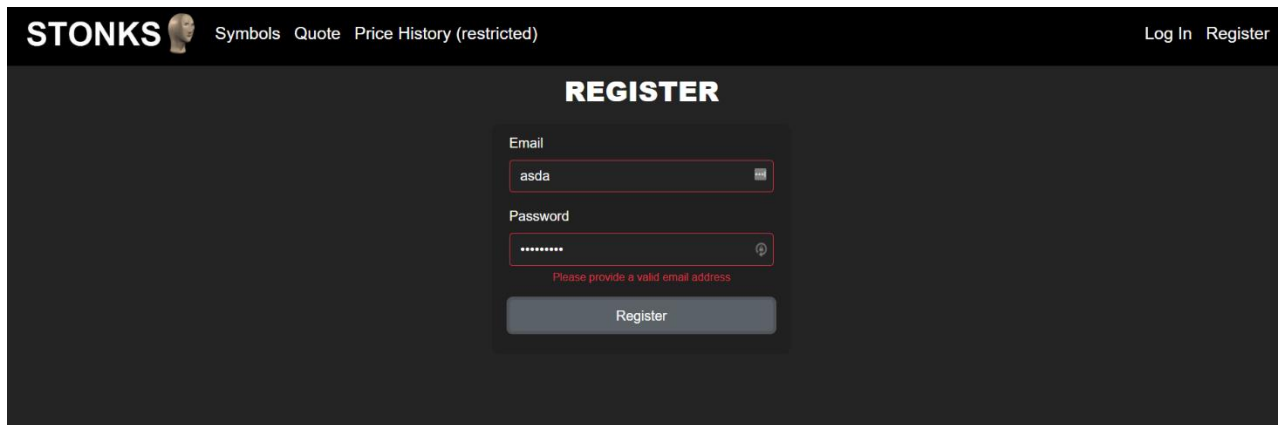
The screenshot shows the 'Price History' page for Agilent Technologies Inc. The page has a dark theme with a navigation bar at the top containing 'STONKS', 'Symbols', 'Quote', and 'Price History', along with a 'Log Out' link. A search bar contains 'A' and a 'Submit' button. Below the search bar is a table with the following data:

Timestamp	Open	High	Low	Close	Volume
26-02-2020	76.68	80.3	76.37	77.43	2879360
27-02-2020	78.68	80.27	78.09	78.11	2528020
28-02-2020	80.7	80.88	77.69	77.96	2788000
29-02-2020	82.22	82.59	81.02	81.5	2918040
24-02-2020	83.9	85.09	83.405	85.07	1785450
21-02-2020	84.8	84.99	83.38	84.54	2481000
20-02-2020	84.7	85.84	83.46	85.38	4741300
19-02-2020	85.68	86.08	84.53	84.78	2827480
18-02-2020	85.72	85.86	85.315	85.82	1962000
17-02-2020	85.3	85.88	84.69	85.47	1598270
16-02-2020	85.4	86.08	85.29	85.47	1484200
13-02-2020	85.82	85.87	84.84	85.59	1542100

Below the table is a chart titled 'Agilent Technologies Inc' showing price movement over time. The chart includes a legend for 'Open Price', 'Close Price', 'High Price', 'Low Price', and 'Volume'.

The authed endpoint is used for the price history tab. If the price history url has query parameters, the price history will display for that symbol, else the form is left blank for users to input a symbol and two dates. The data is displayed in a sortable, filterable table and chart. Unauthorized access to this page is not allowed and the JWT token is checked, but if occurs, no information is displayed. (Image is zoomed out)

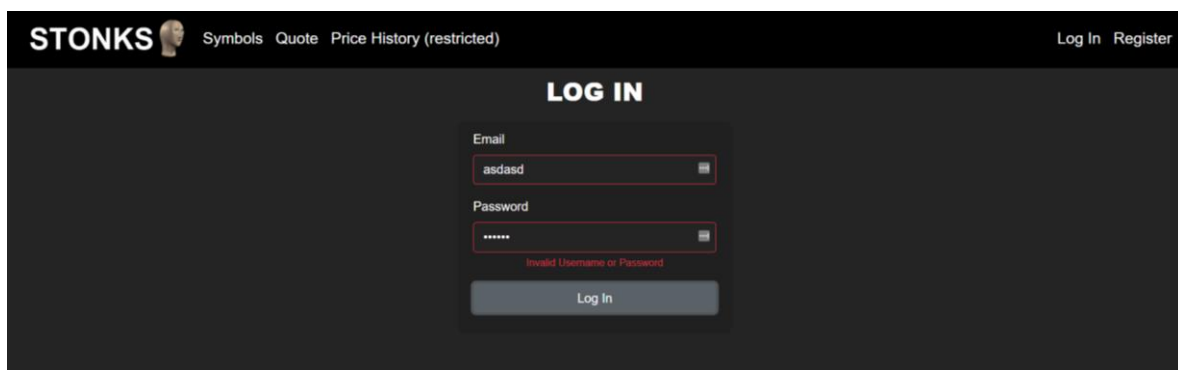
</user/register>



The register contains a form which allows an email and a password. When submitted, it is posted to the database and redirects to the home screen since it automatically logs them in, unless it is invalid in which it gives an appropriate response as to why it didn't go through.

</user/login>

The login contains a form which allows an email and a password. When submitted, it retrieves a JWT token which is stored and then redirects. If the form data is invalid the errors are handled and an appropriate message is given to the user.



Modules used

Ag-grid-react

Module to provide fully-featured table components, including sorting and filtering

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

Bootstrap

Module to provide a CSS Framework to provide styling to the website

<https://getbootstrap.com/>

Reactstrap

Module to provide prebuilt styled form and navigation components, including inputs, buttons and drop downs, built on Bootstrap

<https://reactstrap.github.io/>

Chartjs

Module to provide fully-featured charts in JavaScript, including bar graphs and line graphs

<https://www.chartjs.org/>

React-Chartjs-2

A react wrapper for ChartJs making it easier to use as function based components

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

Chartjs-plugin-zoom

A plugin for Chartjs that adds zoom and pan functionality

<https://github.com/chartjs/chartjs-plugin-zoom>

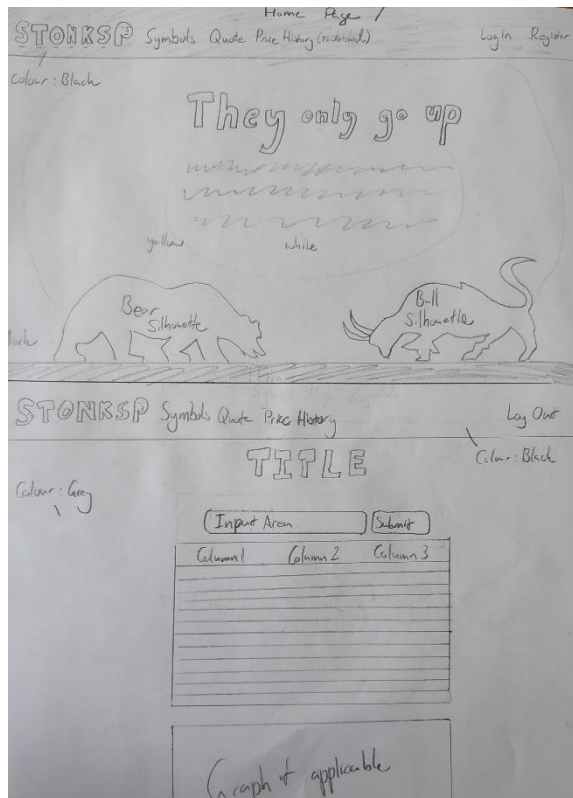
jsonwebtoken

A module for handling JWT's from the database for authentication

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

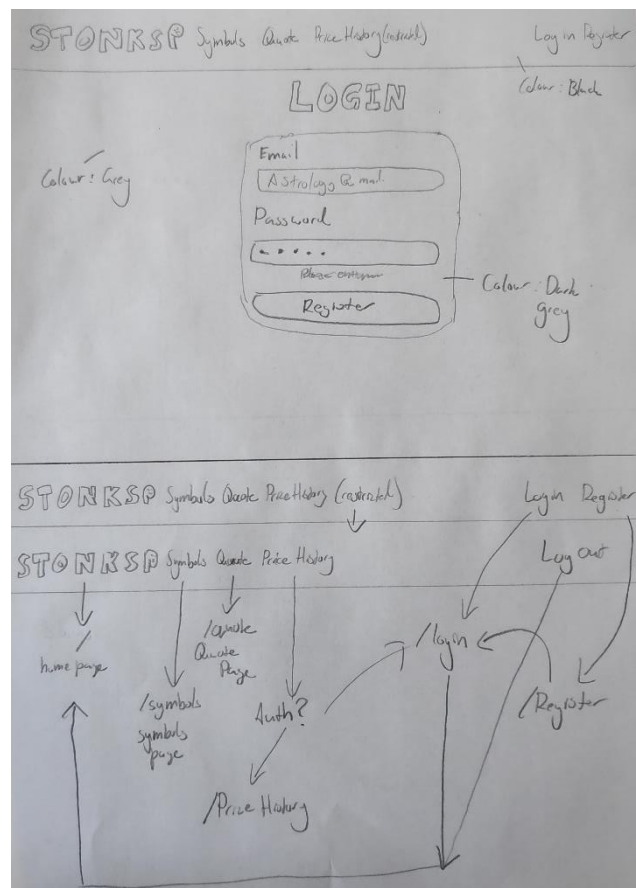
Application Design

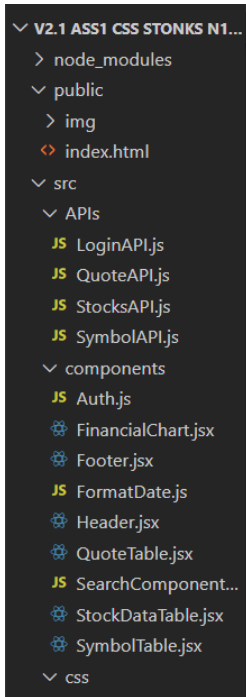
Navigation and Layout



The navigation starts off at the home page "/" and the header bar across the top. This header across the top is used since it is standard for most websites and allows easy access to all pages available in the app. The STONKS and icon provide the link back to the home page, as well as typing any address that isn't recognized by the website. This was chosen since it is standard to have the icon link back to the home page and looks a lot better if instead of a 404, which was considered, it simply returns the website back to home. When the width of the app is restricted, the header bar turns into a hamburger with a drop-down menu offering the same options, which was included so the app retained its form when the tab is dragged. Logging in and/or registering redirects to the home page and the log in, register and price history all change to give notice that the user has logged in. This could be improved by making it go back to the page the user came from which would require handling edge cases like not returning to the register page. The

pictures of how the UI is meant to look are somewhat how it end up being executed, except for the page structure since the register was actively changed from just registering, to also logging in, since it is less clicks for the user. Once the user has chosen a page they will see the layout of the forms. The form design is slightly lacking in terms of aesthetics, with the login/registration forms having duplicated words which doesn't look great. However, placing them without labels makes it look too small for a login page and there was not enough time to properly work with modals. However, functionally they work well, with them being as responsive as any other sites forms. The symbol, quote and price history all have forms which are attempting to be as simple as possible and centered for ease of use, rather than the weird white space left by left-align. The price history from date is slightly annoying since the default upon opening to input a date is the current date, and the data starts 3 months ago. The dark mode style UI was chosen since it is easier on the eye then bright white, especially at night, and saves the users battery (Wilson, 2020).

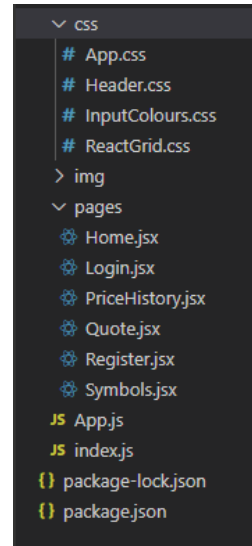




Technical Description

Architecture

The architecture of the application consists of a separated component, API, page layout. The pages are in the pages folder and used in the app.js file which handles the routing. The folder structure consists of the pages, images, apis, css and components. These were chosen as mixing them would be confusing, and there are enough components to warrant separating the apis and css. It would be unwise to place images amongst script files. In times when JSX formatting is helpful, the files are JSX, and when not, they are JS.



The flow of data around the app works via the apis.js exporting the functions to get and post from the server, with the searches and request information being placed in props and given to them. The pages consist of the page, wrapped in a div, and then a function in the same file which makes up the

majority layout and states involved on the page. Components, such as the financial chart are imported and have their data passed into them via props. While the login and register could have their apis be separate files, the overall script size was small enough to warrant keeping all the components together, although the app is designed so they may be spread out if further work increases their size. The CSS files are imported into the appropriate places with files with inputs inputting the input colors. Authorization, formatdate and searchcomponents are used widely across the app and thus are called upon when necessary. The register API sends the login API if the register is complete and uses the login API to do the redirect. All code is commented and formatted in an easy to read way and the file structure allows for simple scope expansion, understanding and editing, making it really easy to edit my existing code and expand upon it.

Test plan

Task	Expected Outcome	Result	Screenshots
Click on STONKS logo	Navigate to home page	PASS	-
Click on Header Links	Navigate to correct page	PASS	-
Access price history while logged out	Redirect to login	PASS	Figure 2
Click on symbol while logged out	Redirect to login	PASS	Figure 2
Shrink page	Hamburger with list appears	PASS	Figure 2
Log in with invalid credentials	Error feedback appears	PASS	Figure 2
Register with invalid email	Error feedback appears	PASS	Figure 1
Register with existing user information	Error feedback appears	PASS	Figure 3
Register with valid information	Directly login	PASS	Figure 4
Login with valid information	Redirected to home	PASS	Figure 4
Search by industry and name in symbols	Returns search information	PASS	Figure 9
Search symbol and date in price history	Returns search information	PASS	Figure 4
Search by symbol in quote	Returns search information	PASS	Figure 10
Search incorrect dates	Error feedback appears	PASS	Figure 5
Search for 1 day/ no date	Return search without chart	PASS	Figure 6
Search for a single date	Return search with boundary	PASS	Figure 4
Search for a bad symbol	Error feedback appears	PASS	Figure 7

Edit chart – zoom, pan, remove line	All work	PASS	Figure 8
Search for quote symbol	Return search	PASS	Figure 9

Difficulties / Exclusions / unresolved & persistent errors /

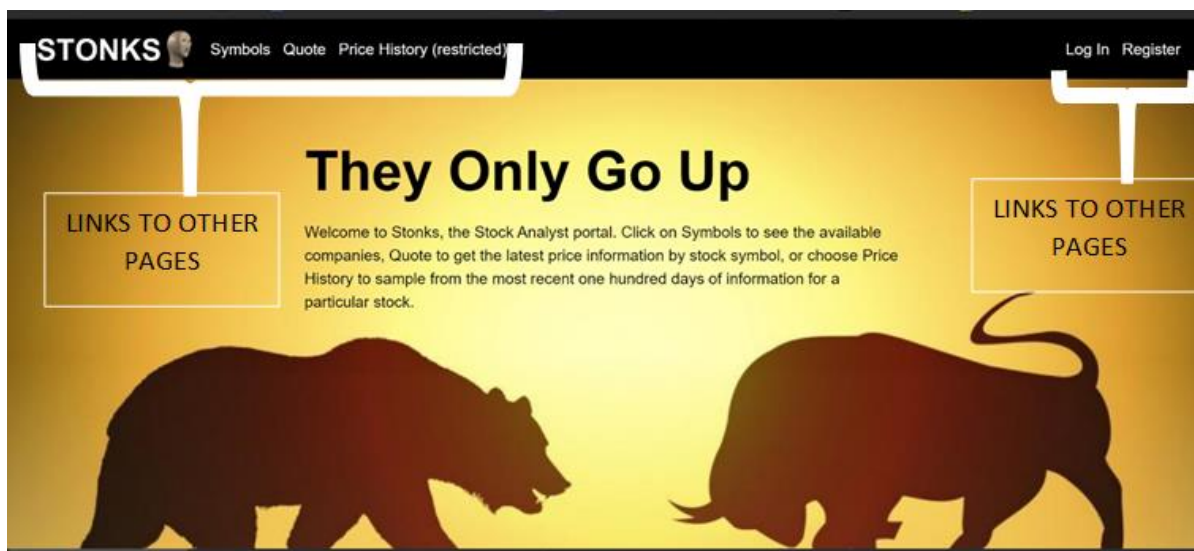
The major roadblocks that I had were dealing with the differences between javascript with React and the previous programming languages. The use of arrow functions using { } and objects using { } repeatedly made my code not work since on occasion, I didn't realize the data I was processing was encapsulated in an object. Once I realized this, I was able to appropriately flow the data from the fetch to the table. Another was the error handling for the project, since no internet connection returns a different error to the ones generated by the API, and the APIs had to be caught twice due to the use of useEffect and my inadequacy. Errors that still persist involve non-unique children and the use of location as an update forcer which is bad practice, but these do not crash or limit the application in anyway.

Extensions

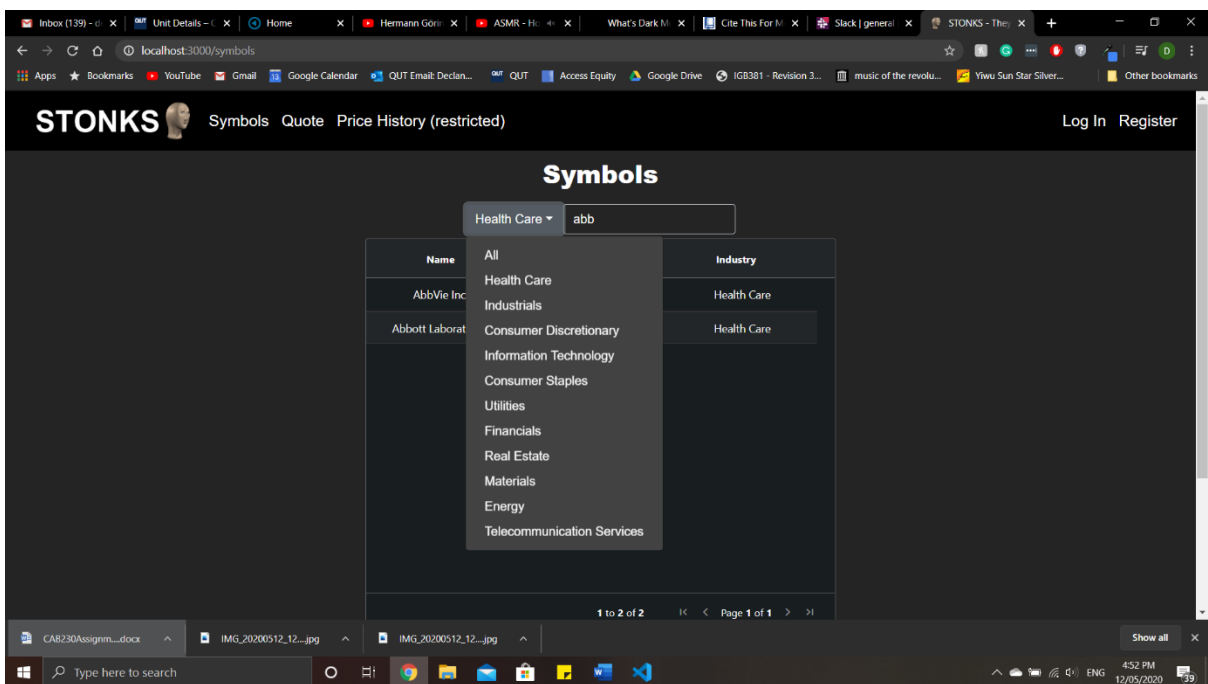
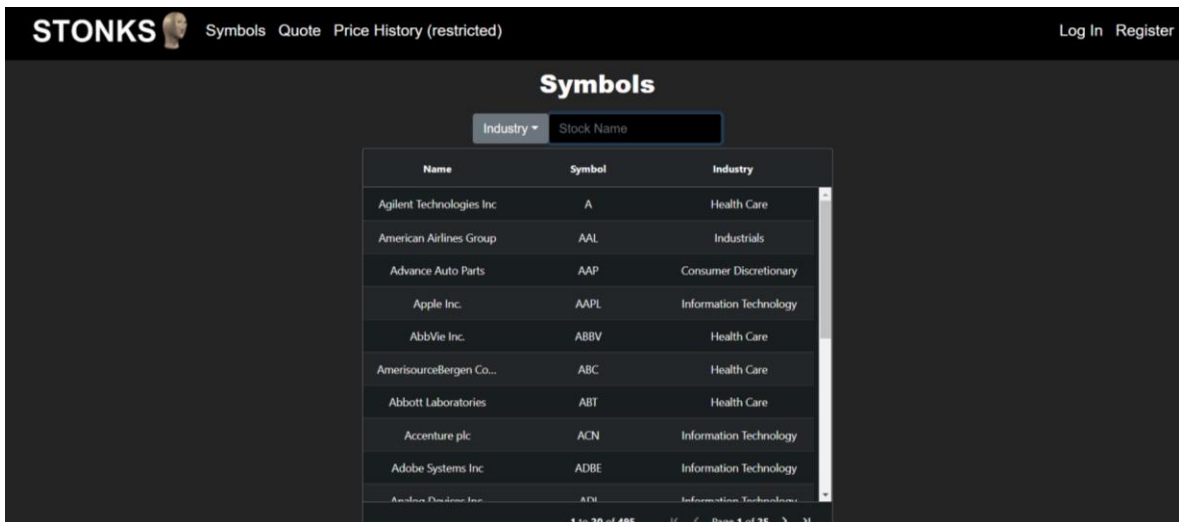
Future extensions for the application would involve cleaning up the UI, expanding the database dataset/ using an official stocks api and adding the functionality to trade stocks. Even further dynamic error handling like with the login would be great for the rest of the application.

User guide

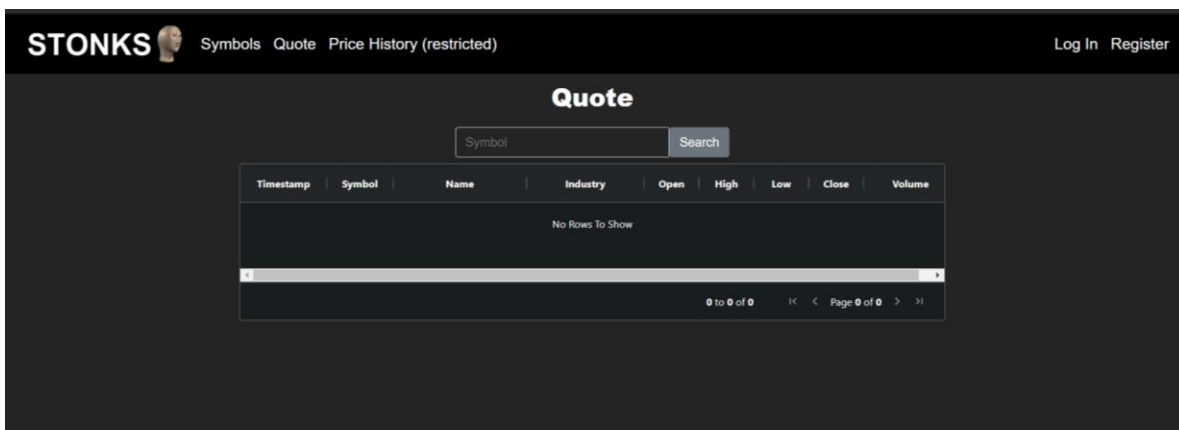
To begin with, you will start off on the home page. For all of the following pages, make sure you are connected to the internet, though if you disconnect, the website will tell you. Across the top will be the header bar which contains links to this home page – via the STONKS logo, the symbol page, quote page, price history page – which will be restricted right now, the log in page and the register page. You may navigate to any one of these pages except the price history page, since it will redirect you to the login page. You may also scroll down to see our helpful tips on the home page.



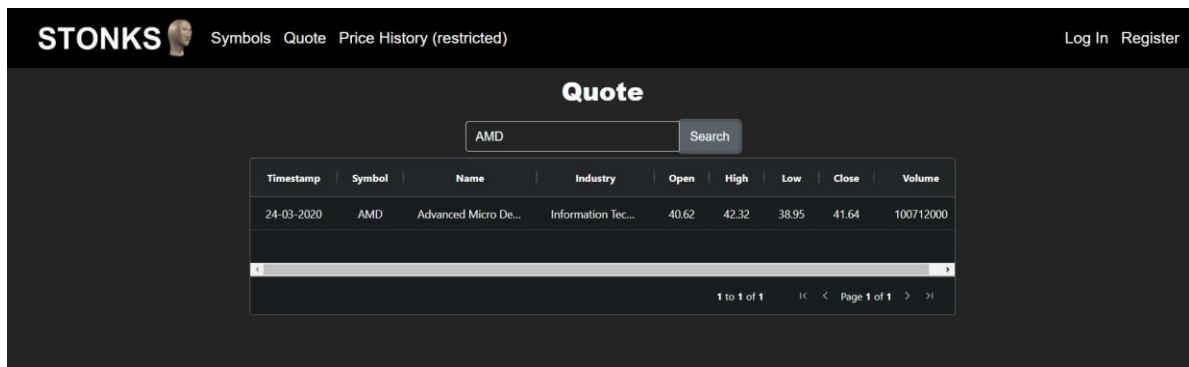
Once you have clicked the symbols tab you will be in the symbols page. From this page you can get a list of symbols that match your search criteria. By default, all industries are selected and no filter is applied. Clicking the dropdown button (with industry on it) will bring up the list of possible industries. Putting anything in the stock name box will filter the data to only bring back the stocks that match. When selecting a new industry the stock name will remain in the box and continue to filter.



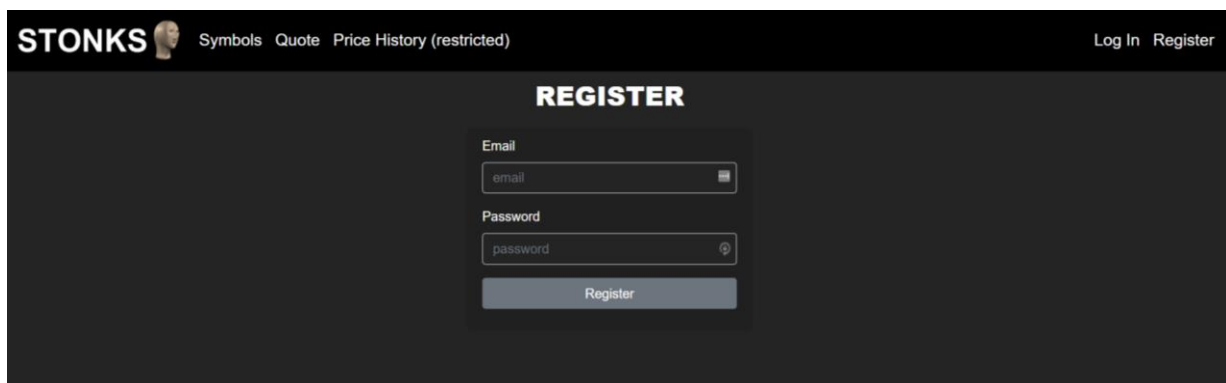
If you select the Quote tab then you will go to the page where you can get a quote. In this tab you can search for a stock symbol. By default, the table is empty.



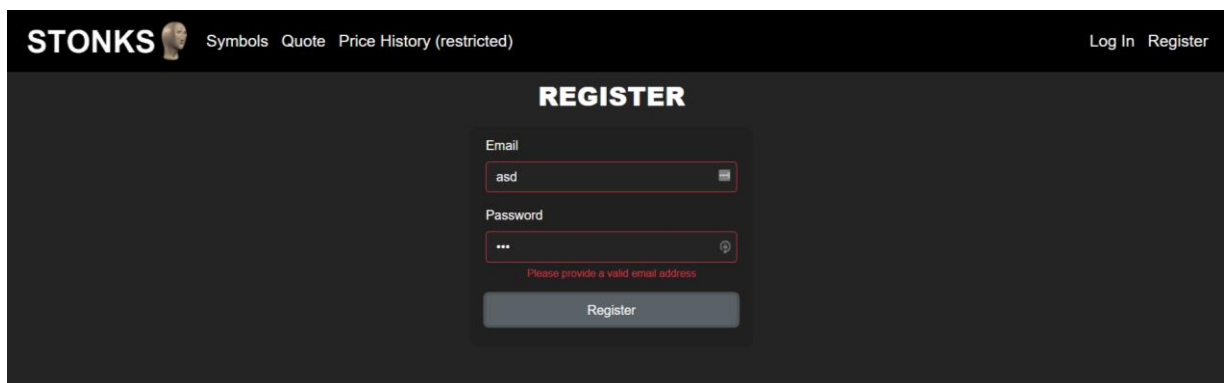
Once you have searched for a symbol then the quote will appear in the table. If you don't enter a valid symbol then the box will tell you that there is a problem and how to address it.



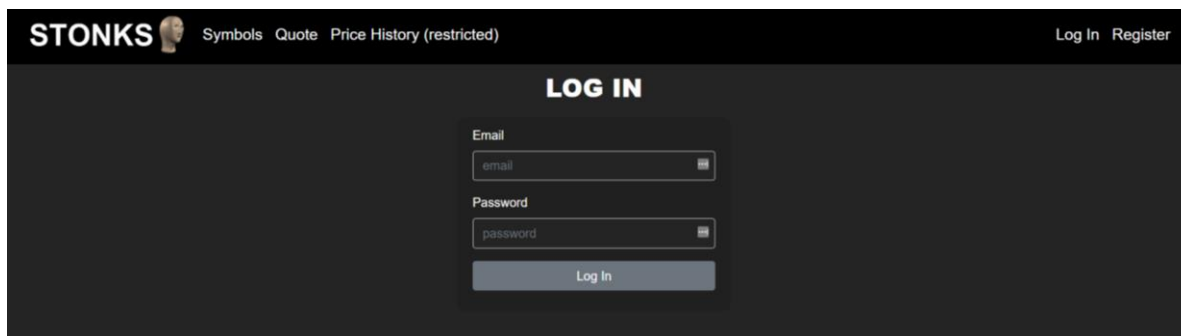
If you click on price history while it is restricted, or the log in button, then you will be directed to the Log In page. To Log in, you need to register with an account. To do this, you can click the register button at the top right and be greeted by the register page.



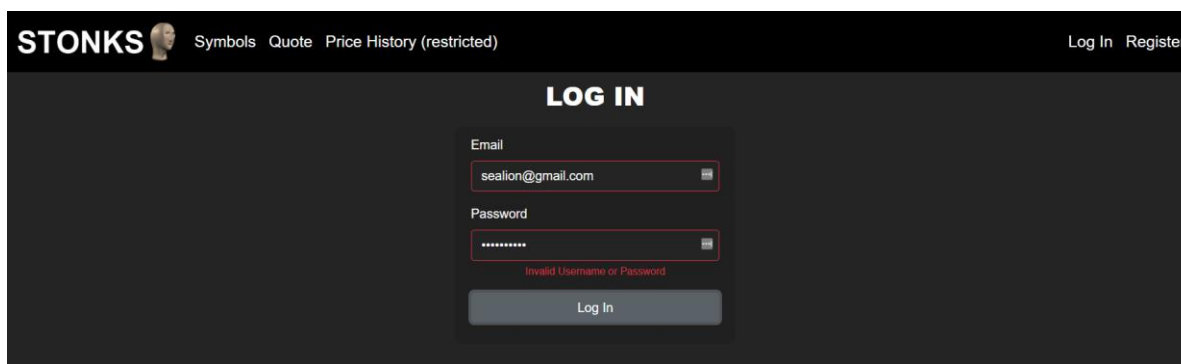
Enter a valid email and password and you will be registered and immediately logged in, going to the home page. If you try to register with an email that isn't an email address, the app will ask you to. If you try to register with an existing account then it will also tell you.



Once you have registered and come back to the site, you should go to the log in page. Here you can log in so that you can access the price history tab.



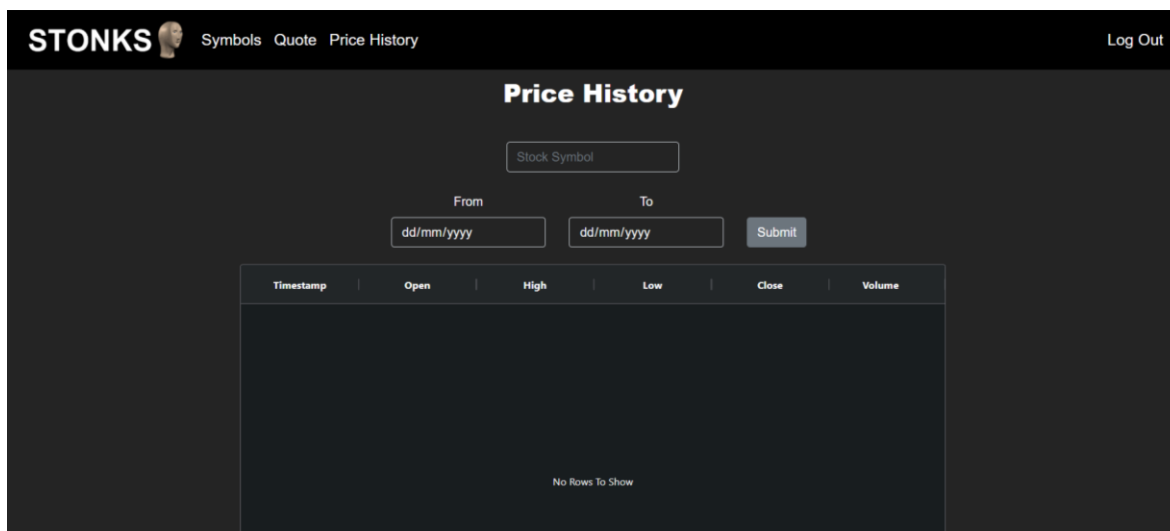
Log in with a registered email and password and if it is correct then it will transport you back to the home page. If you have attempted to log in and your email and password is incorrect, then it will tell you so.



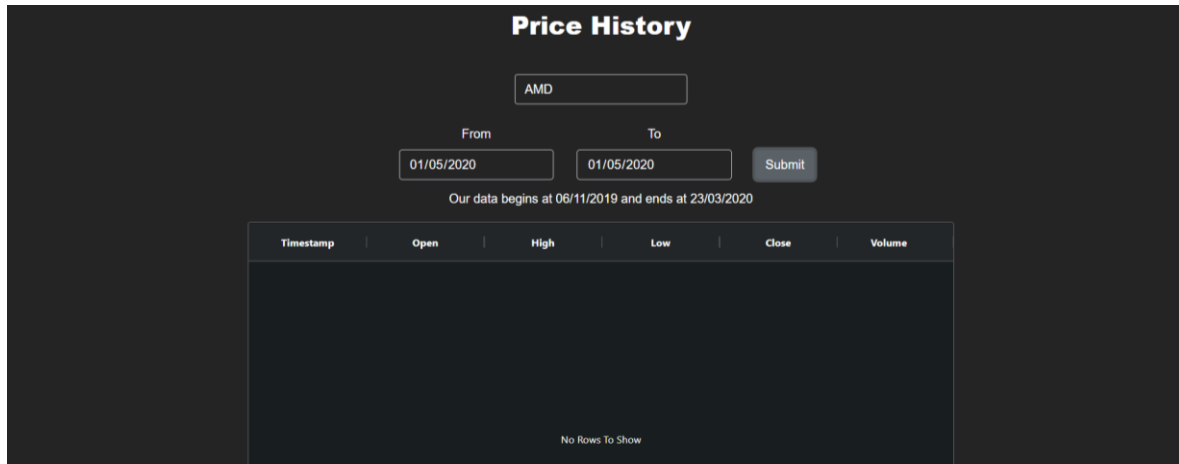
Once you have logged in and be directed to the home page, you will notice that the price history tab no longer says that it is restricted and there is an option to log out.



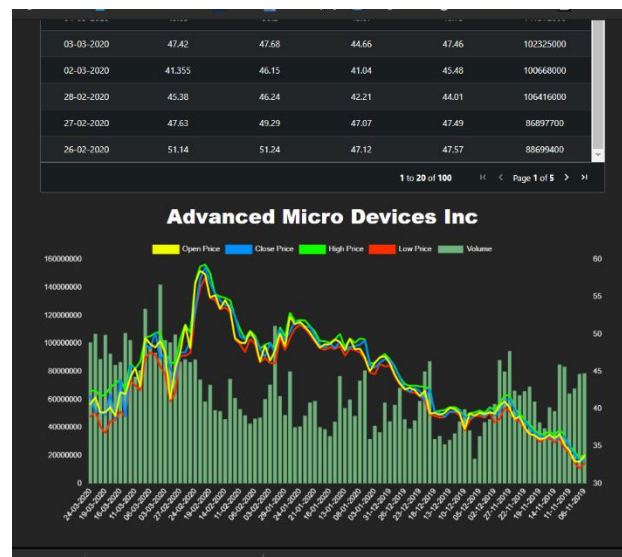
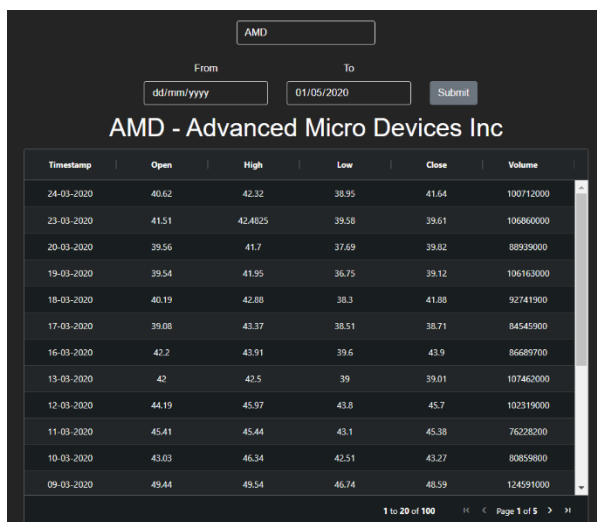
Now you can go to the price history tab or log out. In the price history tab you will see three input boxes. One of these is for a symbol and the other are date boxes, a "TO" and a "FROM".



The Stock Symbol box is for the symbol you wish to know the data for. The from box is the date in which stock data after it will be listed. The to date is the date in which stock will be listed to. Putting none in produces the most current result. Putting in one or the either provides data from or to that date. Putting in both produces data between those dates. If the dates are outside the range or the to is before the from, then no data is shown and an error message will pop up below



If you put in valid dates and a valid symbol, the data will display in the table and in the chart



The data in the table can all be sorted, with price and volume able to be filtered. The chart can have each set of data removed or added by clicking the legend, and can be zoomed in or out. If the user chooses to they can also access the price history tab through the symbols menu by clicking the symbols (only if they are logged in), upon which it will load all data for that stock symbol.

Symbols

Industry: micrd

Name	Symbol	Industry
Advanced Micro Devic...	AMD	Information Technology
Microchip technology	MCHP	Information Technology

Price History

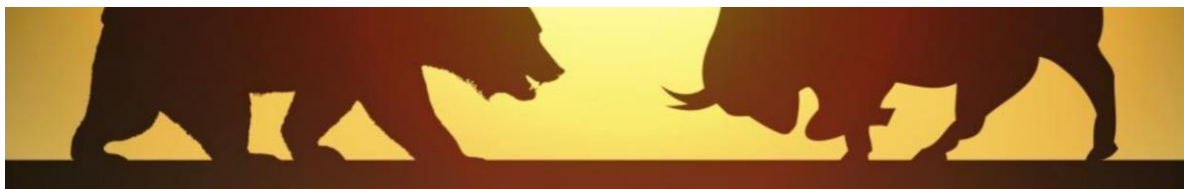
AMD

From: dd/mm/yyyy To: dd/mm/yyyy

AMD - Advanced Micro Devices Inc

Timestamp	Open	High	Low	Close	Volume
23-03-2020	41.51	42.4825	39.58	39.61	106860000
20-03-2020	39.56	41.7	37.69	39.82	88939000
19-03-2020	39.54	41.95	36.75	39.12	106163000
18-03-2020	40.19	42.88	38.3	41.88	92741900
17-03-2020	39.08	43.37	38.51	38.71	84545900
16-03-2020	42.2	43.91	39.6	43.9	86689700

When the user decides to log out, no matter what tab they are in, they will be directed to the home page and the top bar will go back to the way it originally was with price history (restricted), login and register options available. On the home page you can scroll down and see someone helpful site promises.



Our Promise



You will lose sleep. Since these are American stocks you will spend all night randomly checking the market.



The market goes up when it should go down and goes down when it should go up, and the FED can just print infinite money.



Explore welfare options after your life savings disappear within a week, or you lose thousands on options trading.

STONKS
Copyright © 2020

References

Wilson, S. (2020). What's Dark Mode, Anyway?. Retrieved 12 May 2020, from <https://www.braze.com/perspectives/article/whats-dark-mode-anyway>

Appendix

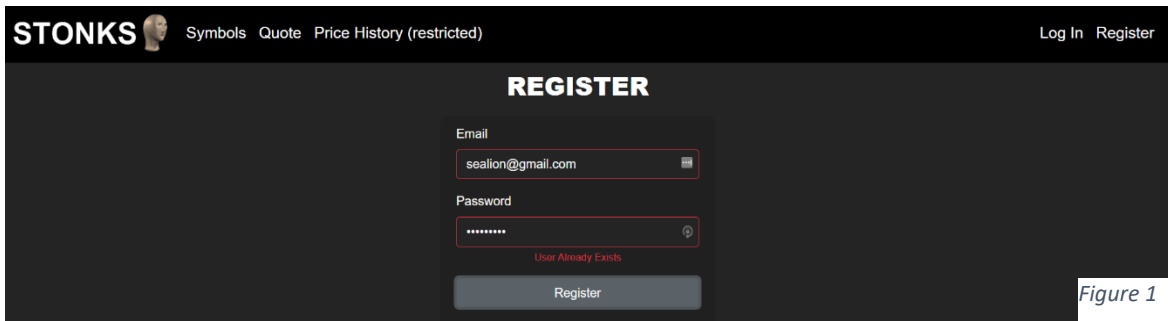


Figure 1

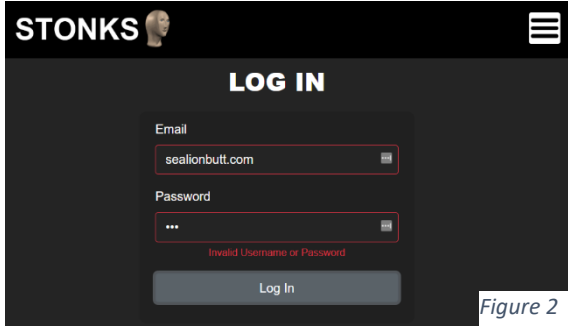


Figure 2

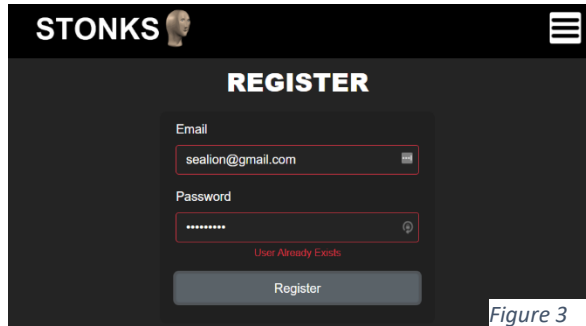


Figure 3



Figure 4

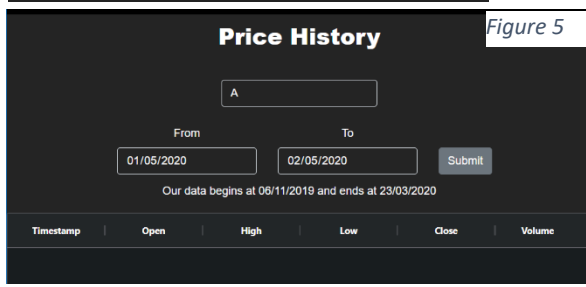


Figure 5

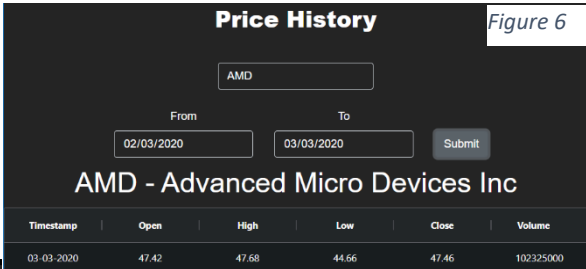


Figure 6

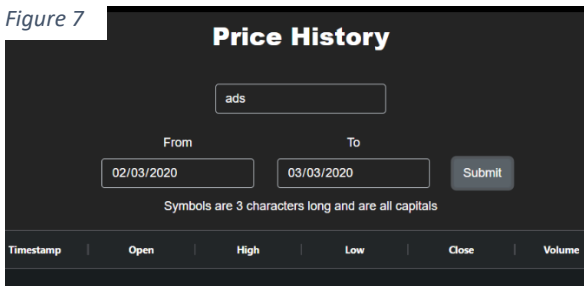


Figure 7

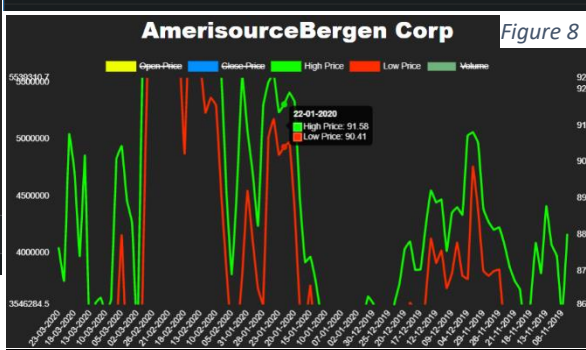


Figure 8

Name	Symbol	Industry
AbbVie Inc.	ABBV	Health Care
Abbott Laboratories	ABT	Health Care

Figure 9

Timestamp	Symbol	Name	Industry	Open	High	Low	Close
24-03-2020	AMD	Advanced Micro De...	Information Tec...	40.62	42.32	38.95	41.64

Figure 10