2021

<Timothy Jeoung/s>

<n10887601/s>

5/9/2021

IFN666 Stocks Web APP – Client Side



IFN666

Stocks Web APP – Client Side

Contents

[Introduction 2](#_Toc71199793)

[Purpose & description 2](#_Toc71199794)

[Completeness and Limitations 2](#_Toc71199795)

[Use of APIs 3](#_Toc71199796)

[Financial Modeling Prep 3](#_Toc71199797)

[Alpha Vantage 3](#_Toc71199798)

[Modules used 4](#_Toc71199799)

[Ag-grid-react 4](#_Toc71199800)

[Axios 4](#_Toc71199801)

[Plotly.js 4](#_Toc71199802)

[Reactstrap 4](#_Toc71199803)

[Application Design 4](#_Toc71199804)

[Navigation and Layout 4](#_Toc71199805)

[Technical Description 5](#_Toc71199806)

[Architecture 5](#_Toc71199807)

[Test plan 6](#_Toc71199808)

[Difficulties / Exclusions / unresolved & persistent errors 7](#_Toc71199809)

[User guide 8](#_Toc71199810)

[Appendices 8](#_Toc71199811)

[Appendix A – self-checking against CRA 8](#_Toc71199812)

[Appendix B – Screenshots of test plan results 9](#_Toc71199813)

## Introduction

### Purpose & description

This client-side web application development allows users to view and analyze stock market data through text, data, tables and charts. It will deliver experience in querying REST APIs, on presenting stocks data for users. Moreover, it will deliver understanding with web technologies such as JavaScript and React.

Through this web application, the users will be able to see all the NASDAQ Stocks information such as their symbol, name and sector. Moreover, it also includes latest 2021 quotes of most popular NASDAQ company such as Apple inc. It can be easily be sorted to required boundary such as highest to lowest or by oldest to newest. This can be done by just clicking on the required subject. Lastly, on Charts section it will show a visualization of a line chart of NASDAQ company closing price in detailed time and date.

### Completeness and Limitations

In this web application it has a simple web page with basic styles that has simple functions such as displaying stocks, symbols, sector and history of most popular NASDAQ company through a line chart for certain period. For chart, *react-plotly* has been used as chart component to visualize detailed line charts. Furthermore, it also uses React and displays the data clearly by using table components with *ag-grid* material. Moreover, it has also implemented pagination and sorting for both ‘Stocks’ and ‘Quote’ tables. Finally, navigation of this web application is handled via *React Router*. However, the only function that is not implemented is search function. Nevertheless, the web application operates sufficiently without search function.

## Use of APIs

### Financial Modeling Prep

Financial Modeling API has been used for “Stocks” to display company name, symbol and sector. Additionally, it has been also used for “Quotes” to display NASDAQ company daily price history.

Screenshot for “Stocks” (Financial Modeling Prep API)Text

Description automatically generated

Screenshot for “Quote” (Financial Modeling Prep API)Text

Description automatically generated

### Alpha Vantage

Alpha Vantage API has been used for “Charts” to display most popular NASDAQ company daily closing price via line chart.

Screenshot for “Charts” (Alpha Vantage API)

Text

Description automatically generated

### Modules used

#### Ag-grid-react

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

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

#### Axios

Component for React, allows to render async API requests to display line chart for web application

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

#### Plotly.js

Module to embed D3 charts in react-powered web application

<https://plotly.com/javascript/react/>

#### Reactstrap

React component for Bootstrap for styling on button and badges.

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

## Application Design

### Navigation and Layout

For this web application, a simple designed has been used for navigation. It has been kept in basic format for users to cleanly navigate between the query and user functions. According to the web application, the navigation bar is displayed on right top section displaying Home, Stocks, Quotes and Charts by clicking on each function it will change pages to desired page. Additionally, React routing menu is applied for routing which can be checked in “App.js”. However, for future use the web application should have more design to attract users to be visualized as professional web application. Currently, the web application is too basic and boring to attract and engage the users.

A picture containing text

Description automatically generated

## Technical Description

### Architecture

Application used:

Graphical user interface, application

Description automatically generated

Src directory:

Graphical user interface, text, application

Description automatically generated

For this web application in src directory, the components and pages were split to be clearly seen and to route easily and efficiently. Component’s folder has “Header.js” for navigation bar, and pages folder have “Home.js”, “Stocks.js”, “Quote.js” and “Chart.js” for 4 pages of the web application.

### Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Expected Outcome** | **Result** | **Screenshot(s)** |
| **Checking home page** | | | |
| Home page shows on http://localhost:3000/ | Home page displayed, with clear navigation | PASS | Figure 1 |
| Check page title is “Stocks” | Page title is “Stocks” | PASS | Figure 2 |
| Filter by name | Stocks filtered by name | PASS | Figure 3 |
| Filter by sector | Stocks filtered by sector | PASS | Figure 4 |
| Quote page shows on http://localhost:3000/Quote | Selected Company quote history displayed clearly | PASS | Figure 5 |
| Check dropdown menu filters by dropdown item selected | Open price sort by numbers and filter less than and greater than | PASS | Figure 6 |
| **Checking the detailed stock information** | | | |
| Detailed stock details shown at http://localhost:3000/Quote | Detailed stock dashboard shown. | PASS | Figure 7 |
| Detailed stock closing price shown at http://localhost:3000/Charts | Detailed line charts shown | PASS | Figure 8 |

Difficulties / Exclusions / unresolved & persistent errors /

During building this web application, the major roadblocks were changing location of navigation bar and fetching data in specific API json format to a column. Location of navigation bar issue has been resolved by a guide of CSS via public resources. This was done by giving a style and margin on each nav function.

Text

Description automatically generated

For fetching, it was hard to fetch the stock quotes in json format. This was solved by adding a keyword ‘historical’ to read and map the json data. By adding ‘historical’ It has fetched the data that was needed to display the stock history.

Text

Description automatically generated

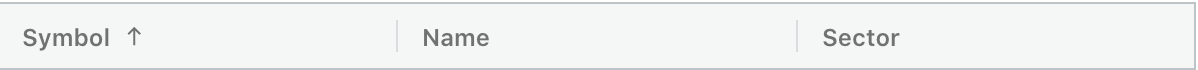
The web functionality that couldn’t be done was viewing different company quotes. The web application was only able to display one company stock history and closing price chart which was Apple inc. (AAPL). To include this functionality the ‘useParams’ has to be used to route and display other NASDAQ company stocks. However, due to lack of knowledge of ‘useParams’ this function wasn’t able to be used. However, hardcoding was done to display other company function, but to make it easier and effective ‘useParams’ is significant in future.

## User guide

1. Click on the desire navigation (right top corner) to view the information of that page



1. For ‘Stocks’ page, to sort the order of the company by alphabetical order click on the columns Symbol, Name or Sector



1. For ‘Quote’ page, users can also sort the order by clicking on columns. For this page it will be sorted in numbers lowest to highest or highest to lowest. Also, filtering is available by clicking on the desired columns right section. Refer to below screenshot.

Graphical user interface

Description automatically generated

1. Clicking on ‘Charts’ page, it will display the closing price of company line charts. As below screenshot the charts can be zoom in and out in more detailed referring to the price, date and time.

Chart, line chart

Description automatically generated

## Appendices

#### Appendix A – self-checking against CRA

|  |  |  |
| --- | --- | --- |
| Marks | Grade level (in 1- 7 scale) my work belongs to (delete the ones not suitable) | Marks I think I should get |
| **Overall Functionality**  **(30 marks)** | 5 - 6 | 20 out of 30 |
| **Application performance**  Note: balance between client and server processing  **(10 marks)** | . 6 | 8 out of 10 |
| **Application Robustness**  **(10 marks)** | 7 | 10 out of 10 |
| **Application UI Design**  Note: this is purely based on function and usability. There is no direct assessment of the quality of the graphic design.  **(10 marks)** | 5 | 7 out of 10 |
| **Application architecture and Code Quality**  **(10 marks)** | 5-6 | 7 out of 10 |
|  |  |  |
| **Report and Reflections**  **(20 marks)** | 6 | 15 out of 20 |
|  |  |  |
| **Video demo**  **(10 marks)** | 7 | 10 out of 10 |
|  |  |  |
| **Overall Marks (100)** |  | xx out of 100 |

#### [Appendix B – Screenshots of test plan results](#_Toc22385)

***Figure 1***

Graphical user interface, text, email

Description automatically generated

***Figure 2***

Graphical user interface, text

Description automatically generated

***Figure 3***

Table

Description automatically generated

***Figure 4***

Graphical user interface, application

Description automatically generated

***Figure 5***

***Table

Description automatically generated***

***Figure 6***

***Graphical user interface

Description automatically generated***

***Figure 7***

***Table

Description automatically generated***

***Figure 8***

***Graphical user interface, chart, application

Description automatically generated***