

# 2021

## IFN666 Stocks Web APP – Client Side



IFN666

Stocks Web APP – Client Side

<Pattarachai Roongsritong>

<n10548467>

5/7/2021

## Contents

Introduction .....	2
Purpose & description .....	2
Completeness and Limitations.....	3
Use of APIs .....	3
Financial Modelling Prep .....	3
Use of End Points of the chosen Stock API .....	3
All Nasdaq 100 constituents .....	3
Historical Daily Prices with change and volume Time series .....	3
Modules used .....	4
Ag-grid-react.....	4
react-router-dom.....	4
react-chartjs-2 .....	4
react-datepicker .....	4
reactstrap.....	4
bootstrap .....	4
Application Design .....	5
Navigation and Layout.....	5
Technical Description .....	5
Architecture .....	5
Test plan .....	7
Difficulties / Exclusions / unresolved & persistent errors.....	8
Extensions (Optional).....	8
User guide.....	8
Appendices .....	9
Appendix A – self-checking against CRA .....	9
Appendix B – Screenshots of test plan results.....	10

## Introduction

### Purpose & description

This web application is called ‘Stock Monitor’ which allows users to conveniently retrieve stock data of the 102 largest listing companies in the NASDAQ market (NASDAQ-100). The functionalities include in the app are searching by a stock symbol, searching by stock sector, selecting start date to look at more details of each stock such as closing prices which are visualised with a line chart and opening prices. Additionally, the web site is responsive so that it can be accessed via multiple platforms.

This app utilises Hooks on React components as a state management which allows to store response data from a request and avoid reaching out to a server frequently. Apart from this, React Router module is also used to give users smooth browsing experience without refreshing the whole web page. React strap makes the web have nice feel and touch as well as Chart JS and Ag-grid.

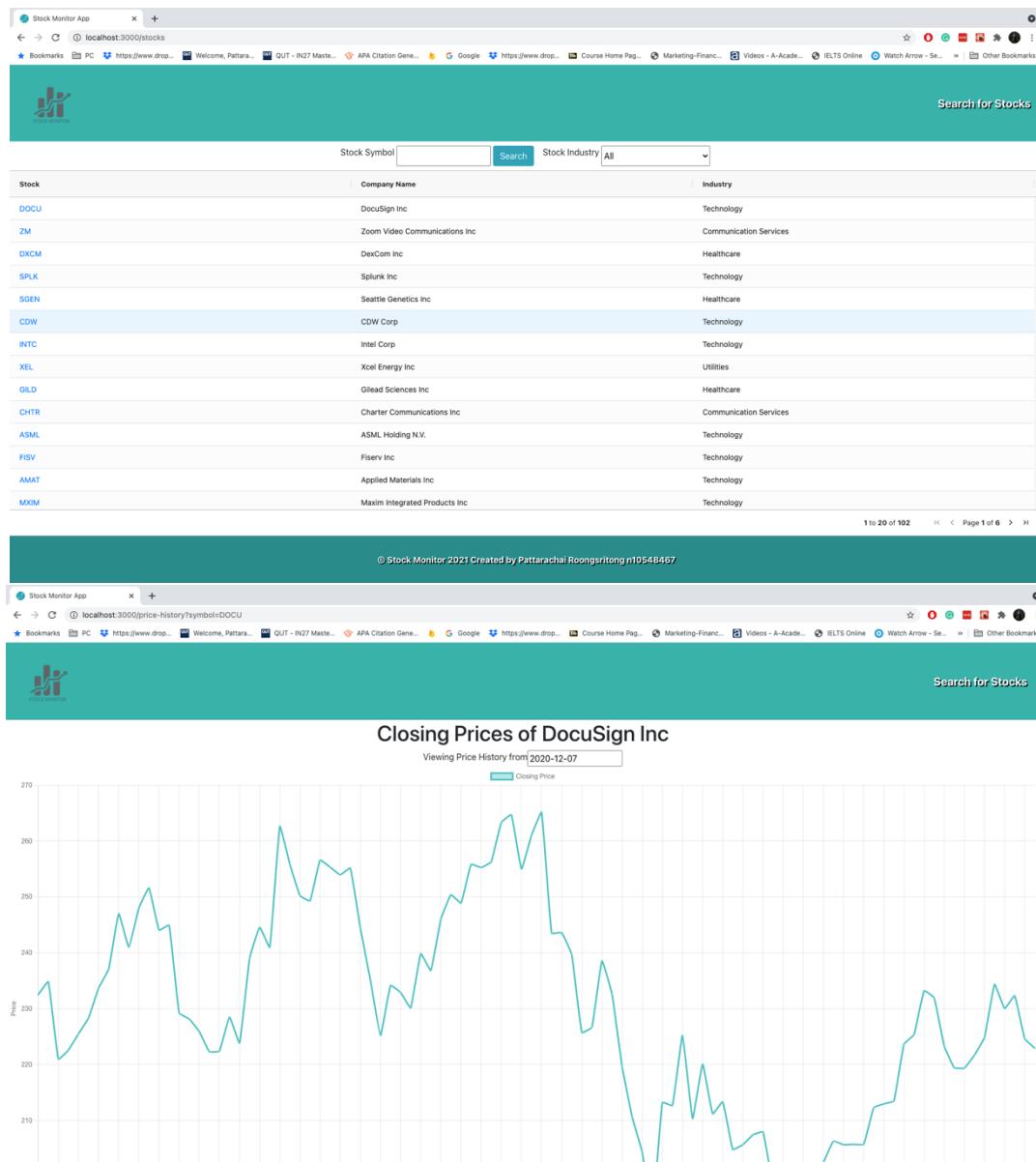


Figure 1. Overview of the web application

## Completeness and Limitations

The main features of this web application are implemented in the way that aims to grade 6 – 7. There are displaying stock in a table using Ag-grid table component which can be sorted and has pagination of 20 items, search bar for stock symbol, search by stock industry with a dropdown list, 100 days or less (adjustable) historical price details of each stock in an Ag-grid table. The useState hook is used to store responses from a server and manipulate the data within a single request from a server. React Router is also used as a navigation to avoid reloading an entire page and visualisation of the closing price is done by using ChartJS. Furthermore, the web app is responsive to multiple devices.

## Use of APIs

### *Financial Modelling Prep*

#### Use of End Points of the chosen Stock API

##### *All Nasdaq 100 constituents*

It gives the data of 102 largest companies in the NASDAQ market including details such as symbol, name, sector, subsector, and headquarter.

The screenshot shows a network request labeled "RESPONSE". The URL is [https://financialmodelingprep.com/api/v3/nasdaq\\_constituent?apikey=demo](https://financialmodelingprep.com/api/v3/nasdaq_constituent?apikey=demo). The response body contains JSON data for a single company:

```
https://financialmodelingprep.com/api/v3/nasdaq_constituent?  
apikey=demo  
[ {  
    "symbol" : "ZM",  
    "name" : "Zoom Video Communications Inc",  
    "sector" : "Communication Services",  
    "subSector" : "Communication Services",  
    "headQuarter" : "San Jose, CALIFORNIA",  
    "dateFirstAdded" : "2019-04-18",  
    "cik" : "0001585521",  
    "founded" : "2019-04-18"  
}
```

Figure 2. All Nasdaq 100 constituent endpoint

##### *Historical Daily Prices with change and volume Time series*

This endpoint gives daily time series stock historical prices with change and volume for the last x days.

The screenshot shows a network request for historical daily price data for AAPL. The response body contains JSON data for a single day:

```
{  
  symbol: "AAPL",  
  historical: [  
    {  
      date: "2021-04-30",  
      open: 131.78,  
      high: 133.56,  
      low: 131.07,  
      close: 131.46,  
      adjClose: 131.46,  
      volume: 106404233,  
      unadjustedVolume: 106404233,  
      change: -0.32,  
      changePercent: -0.243,  
      vwap: 132.03,  
      label: "April 30, 21",  
      changeOverTime: -0.00243  
    },  
    ...  
  ]  
}
```

Figure 3. Historical daily price endpoint

Modules used

*Ag-grid-react*

The module provides a fully featured table component, including sorting and filtering.

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

*react-router-dom*

The module allows us to navigate between components and pages without reloading the entire page.

<https://reactrouter.com/>

*react-chartjs-2*

The module enables us to use Chart.js in a form of React component.

<https://reactchartjs.github.io/react-chartjs-2/#/>

*react-datepicker*

The module is for a date picker component in React.

<https://reactdatepicker.com/>

*reactstrap*

The module allows us to leverage the power of Bootstrap specifically for React components.

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

*bootstrap*

The module is for utilising Bootstrap utilities to style the app for responsiveness without writing HTML, CSS, and JavaScript from scratch.

<https://getbootstrap.com/>

## Application Design

### Navigation and Layout

The application is composed with three pages. The first page is a landing page which intends to be a welcoming point and allows users to navigate to the main features of the app via the navigation bar. The second page is a stock search page. Initially, the table is filled with all NASDAQ-100 companies including their symbols, company name, and industries. The users can type a symbol to search for a stock in the search box as well as filter by an industry by choosing from a drop-down list. Moreover, users can click any stock symbol in the table to navigate to historical price details of the company. This will open a new page showing a line chart for closing prices and table of price details. Users can also select a date to start showing prices until the present which limit to 100 recent days or less. Additionally, users can always come back to the landing page by clicking the logo of this app.

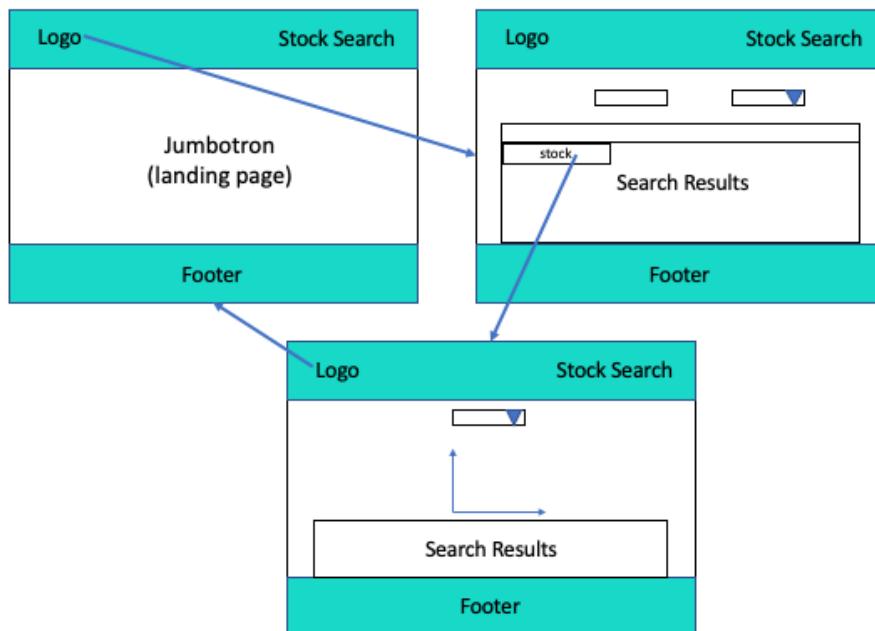


Figure 3. Layout of the web application

The application flow is easy to use; however, when filtering stocks, if there is no stock to show there is only a message from the table stating that no rows to show which can make users confused because when fetching the initial data with a slow internet connection it also shows the same message and the date picker just allows users to select the starting date to show closing prices which might be more counter-intuitive than choosing starting and ending dates as a range to display.

### Technical Description

#### Architecture

The source code of the application is structured by create-react-app. There are three main folders which are `node_modules`, `public`, and `src`. The `public` folder was used to change the title and the icon of browser tab when opening the application. For the `node_modules`, it contains dependency files that were installed in the application and the listed of the dependencies can be found in `package.json` file. The `src` folder mainly contains the React JavaScript code for the project.

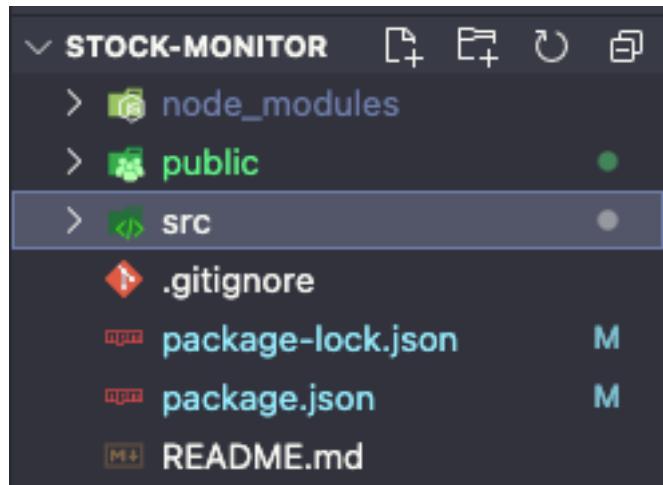


Figure 4. Application skeleton

In the src folder, the app component is an entry point to other components. Multiple folders were created to store components of each page and wrapped in a folder called ‘components’. Home is for the landing page, price\_history is for a historical price detail page, and stock is for stock search page. Additionally, layout folder was created to store reusable components in every page which are navigation bar and footer.

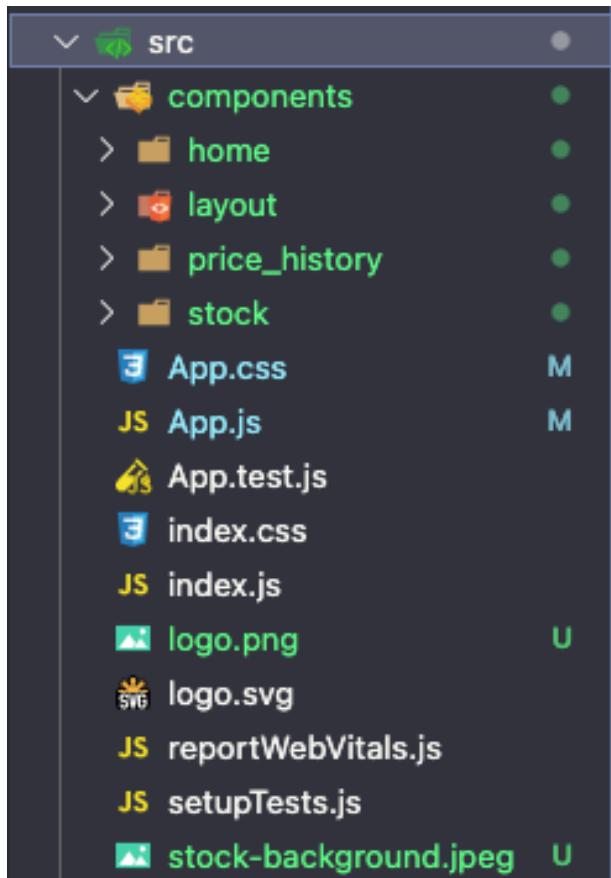


Figure 5. SRC folder organisation

Test plan

Task	Expected Outcome	Result	Screenshot(s)
<b>Checking landing page</b>			
Landing page shows on <a href="http://localhost:3000/">http://localhost:3000/</a>	Landing page displayed and title says Stock Monitor App with a logo	PASS	Figure 6
<b>Checking stock search page</b>			
Stock search page shows on <a href="http://localhost:3000/stocks">http://localhost:3000/stocks</a>	Stock search page displayed	PASS	Figure 7
Check page the total number of stocks	There are 102 stocks	PASS	Figure 7
Type into Stock Symbol to filter by symbol	Stocks filtered by name	PASS	Figure 8
Select Stock Industry to filter by industry	Stocks filtered by industry	PASS	Figure 9
Type into Stock Symbol and select Industry to filter by both	Stocks filtered by both name and industry	PASS	Figure 10
Search by "invalid" symbol	Table shows a message 'No rows to show'	PASS	Figure 11
Search by "invalid" industry	Table shows a message 'No rows to show'	PASS	Figure 12
Click the logo to go back to the landing page	Navigate back to the landing page	PASS	-
Sort each column in the table head	The items in table are sorted alphabetically	PASS	Figure 13
<b>Checking historical price details</b>			
Detailed stock details shown at <a href="http://localhost:3000/price-history?symbol={Stock Symbol}">http://localhost:3000/price-history?symbol={Stock Symbol}</a>	Closing price line chart and table with details shown	PASS	Figure 14
Select start date within 100 days	The chart and table show according to the date range	PASS	Figure 15
Select start date more than 100 days	The chart and table show details for 100 days	PASS	Figure 16
Select start date in the future	Nothing in the chart and table shown	PASS	Figure 17
Click the logo to go back to the landing page	Navigate back to the landing page	PASS	-
Click Search for Stocks to go back to the stock search page	Navigate to the stock search page	PASS	-
Sort each column in the table head	The items in table are sorted numerically and by date for the date column	PASS	Figure 18

## Difficulties / Exclusions / unresolved & persistent errors /

Developing a modern web application involves several technologies making it difficult to grasp and overwhelming to pick up as a beginner. The provided resources alone do not enough to get the grade that I desire, as a result, I had to get through external online tutorials and tried to read documentations by myself. I think self-learning is crucial to be a great developer, but my journey was frustrated and demanding since the submission deadline is very tight. On top of that, I did what I could to interact with classmate but studying online in different time zones makes it even more tough to share resources and form a study group with peers.

## Extensions (Optional)

More stock data in different markets can be added in the application as well as the authentication feature to allow only members to use the service and keep tracking data for product development such as equipping Google Analytics.

## User guide

For local setup, it requires node package manage (NPM). Run command npm install and npm start. After that, navigate to <http://localhost:3000> and the landing page should be displayed. Start viewing the complete list of stocks by clicking 'Search for Stocks' on the top right corner. The information displayed in the table can be filtered by typing in a stock symbol or choosing an industry sector from the dropdown list. More detail of a stock can be retrieved by clicking a stock symbol on a table row and it should display a new page with a line graph with a detail table of daily price information of recent 100 days. Start date can be selected from a date picker to view the details in different date range. A video guide can be found via this link <https://youtu.be/PBilACS3vq8>.

## 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	
<b>Overall Functionality (30 marks)</b>	7	30	out of 30
<b>Application performance</b> Note: balance between client and server processing <b>(10 marks)</b>	7	10	out of 10
<b>Application Robustness (10 marks)</b>	7	10	out of 10
<b>Application UI Design</b> Note: this is purely based on function and usability. There is no direct assessment of the quality of the graphic design. <b>(10 marks)</b>	5	6	out of 10
<b>Application architecture and Code Quality (10 marks)</b>	5	6	out of 10
<b>Report and Reflections (20 marks)</b>	6	14	out of 20
<b>Video demo (10 marks)</b>	7	9	out of 10
<b>Overall Marks (100)</b>		85	out of 100

## Appendix B – Screenshots of test plan results

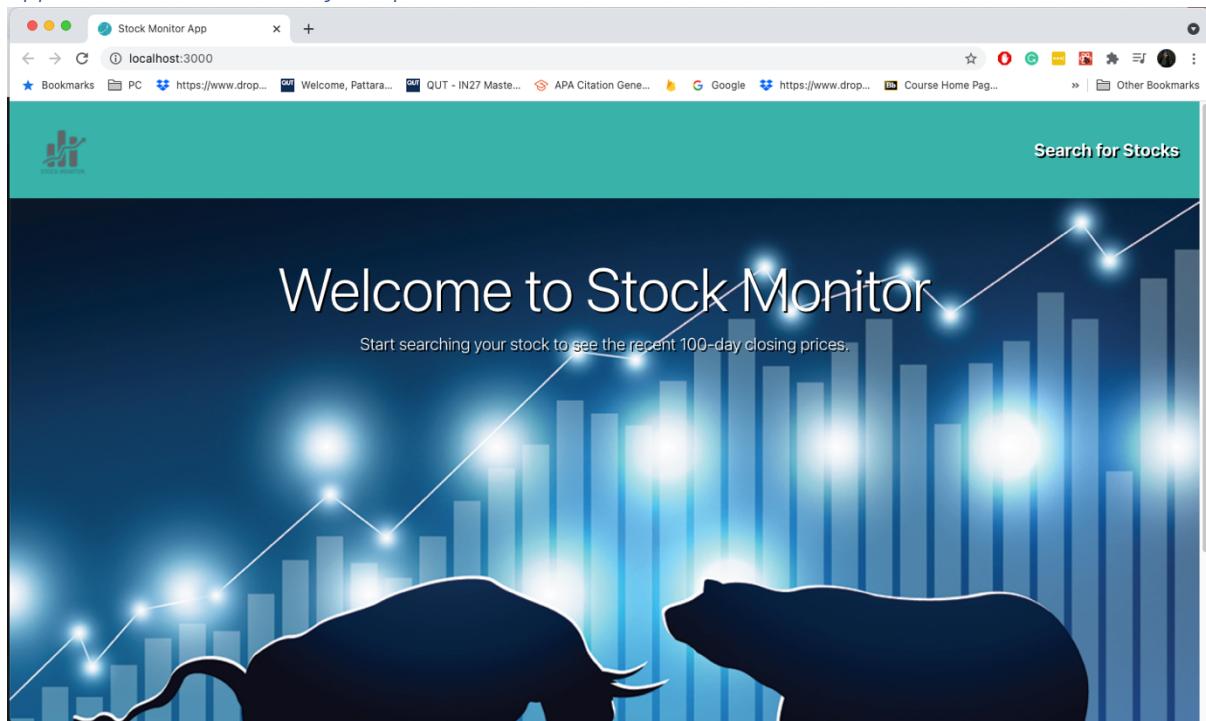


Figure 6. Displaying a landing page

A screenshot of a web browser window titled "Stock Monitor App" showing the search results page. The page has a teal header with the title "Search for Stocks". Below the header is a search form with fields for "Stock Symbol" and "Stock Industry" (set to "All"). The main content area displays a table of stock information. The table has columns for "Stock", "Company Name", and "Industry". The data is paginated at the bottom with links for "1 to 20 of 102" and navigation arrows.

Figure 7. Showing stock search page

The screenshot shows a web browser window titled "Stock Monitor App" with the URL "localhost:3000/stocks". The page has a teal header with the title "Search for Stocks" and a logo of a stylized person. Below the header is a search bar with "Stock Symbol" input field containing "INTU", a "Search" button, and "Stock Industry" dropdown set to "All". The main content area displays a table with three columns: "Stock", "Company Name", and "Industry". A single row is shown for "INTU", which corresponds to "Intuit Inc." under "Company Name" and "Technology" under "Industry". At the bottom of the page, there is a footer bar with the text "© Stock Monitor 2021 Created by Pattrachai Roongsritong n10548467".

Stock	Company Name	Industry
INTU	Intuit Inc.	Technology

Figure 8. Search a stock

The screenshot shows a web browser window titled "Stock Monitor App" with the URL "localhost:3000/stocks". The page has a teal header with the title "Search for Stocks" and a logo of a stylized person. Below the header is a search bar with "Stock Symbol" input field empty, a "Search" button, and "Stock Industry" dropdown set to "Technology". The main content area displays a table with three columns: "Stock", "Company Name", and "Industry". Multiple rows are listed, all belonging to the "Technology" industry. The companies listed include DocuSign Inc., Splunk Inc., CDW Corp., Intel Corp., ASML Holding N.V., Fiserv Inc., Applied Materials Inc., Maxim Integrated Products Inc., Micron Technology Inc., and Cognizant Technology Solutions Corp. At the bottom of the page, there is a footer bar with the text "© Stock Monitor 2021 Created by Pattrachai Roongsritong n10548467".

Stock	Company Name	Industry
DOCU	DocuSign Inc	Technology
SPLK	Splunk Inc	Technology
CDW	CDW Corp	Technology
INTC	Intel Corp	Technology
ASML	ASML Holding N.V.	Technology
FISV	Fiserv Inc	Technology
AMAT	Applied Materials Inc	Technology
MIMX	Maxim Integrated Products Inc	Technology
MU	Micron Technology Inc	Technology
CTSH	Cognizant Technology Solutions Corp	Technology

Figure 9. Search by industry sector

Stock Monitor App    localhost:3000/stocks

Search for Stocks

Stock	Company Name	Industry
ASML	ASML Holding N.V.	Technology
AMAT	Applied Materials Inc	Technology
AAPL	Apple Inc	Technology
ADBE	Adobe Inc	Technology
ANSS	ANSYS Inc	Technology
AMD	Advanced Micro Devices Inc	Technology
WDAY	Workday Inc	Technology
TEAM	Atlassian Corporation PLC	Technology
ADI	Analog Devices Inc	Technology
AVGO	Broadcom Inc	Technology

1 to 14 of 14    Page 1 of 1

© Stock Monitor 2021 Created by Pattarachai Roongsritong n10548467

Figure 10. Search by stock symbol and industry

Stock Monitor App    localhost:3000/stocks

Search for Stocks

Stock	Company Name	Industry
aaaa		All

No Rows To Show

0 to 0 of 0    Page 0 of 0

© Stock Monitor 2021 Created by Pattarachai Roongsritong n10548467

Figure 11. Search by invalid symbol

The screenshot shows a web browser window titled "Stock Monitor App" at the URL "localhost:3000/stocks". The main title is "Search for Stocks". There are two search input fields: "Stock Symbol" and "Stock Industry". The "Stock Industry" field is set to "Basic Materials". Below the search bar is a table header with columns: "Stock", "Company Name", and "Industry". A message "No Rows To Show" is displayed in the center of the table area. At the bottom of the page, there is a footer bar with the text "© Stock Monitor 2021 Created by Pattarachai Roongsritong n10548467".

Figure 12. Search by invalid industry

The screenshot shows a web browser window titled "Stock Monitor App" at the URL "localhost:3000/stocks". The main title is "Search for Stocks". There are two search input fields: "Stock Symbol" and "Stock Industry". The "Stock Industry" field is set to "All". Below the search bar is a table. The first column is "Stock" with an upward arrow indicating it is sorted. The table lists 102 companies, each with its stock symbol, company name, and industry. The industries listed include Technology, Technology, Technology, Industrials, Technology, Utilities, Healthcare, Healthcare, Technology, and Technology. The last row, "AMD", is highlighted with a light blue background. At the bottom of the page, there is a footer bar with the text "© Stock Monitor 2021 Created by Pattarachai Roongsritong n10548467".

Figure 13. Sort a column

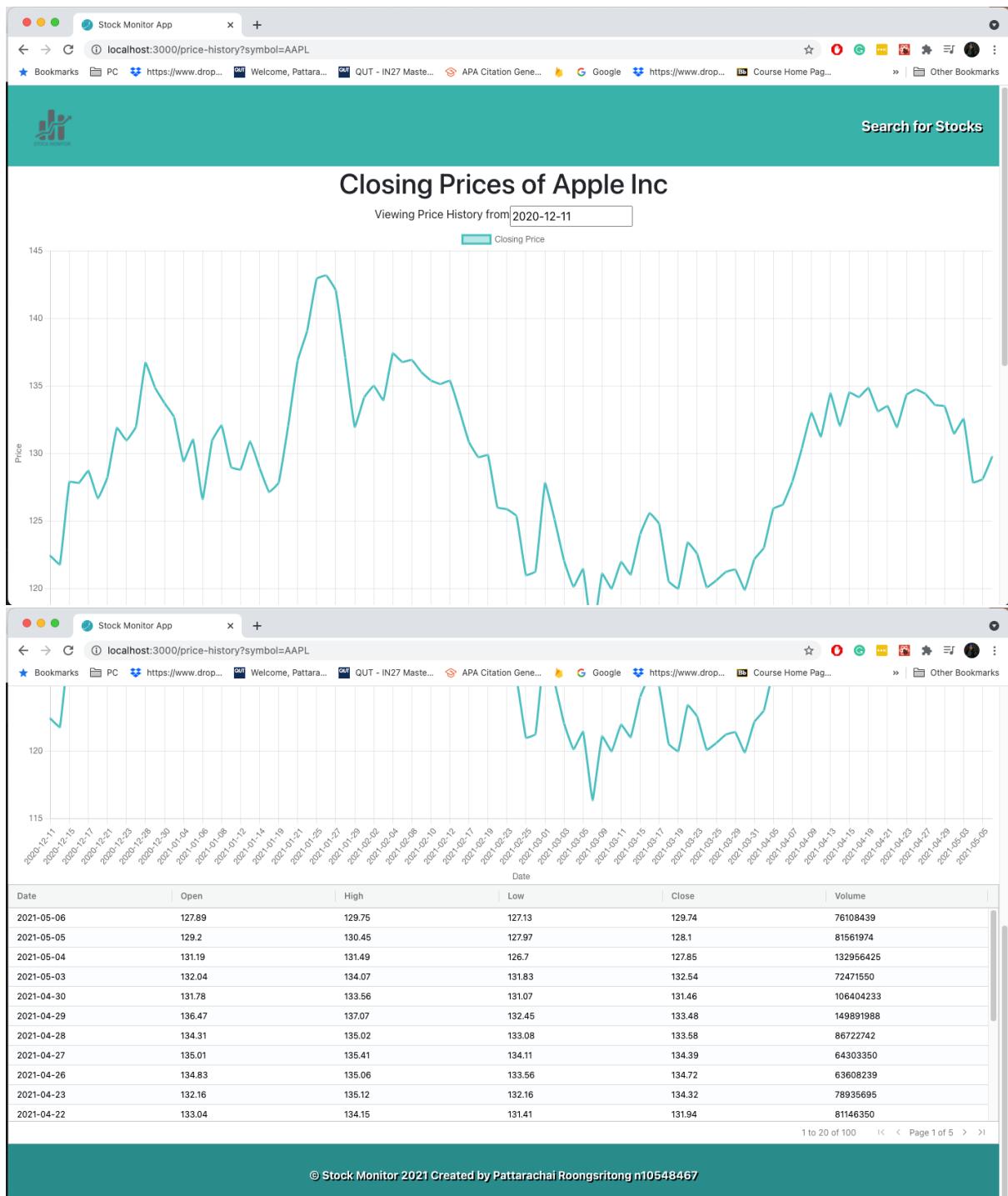


Figure 14. Displaying stock price detail page

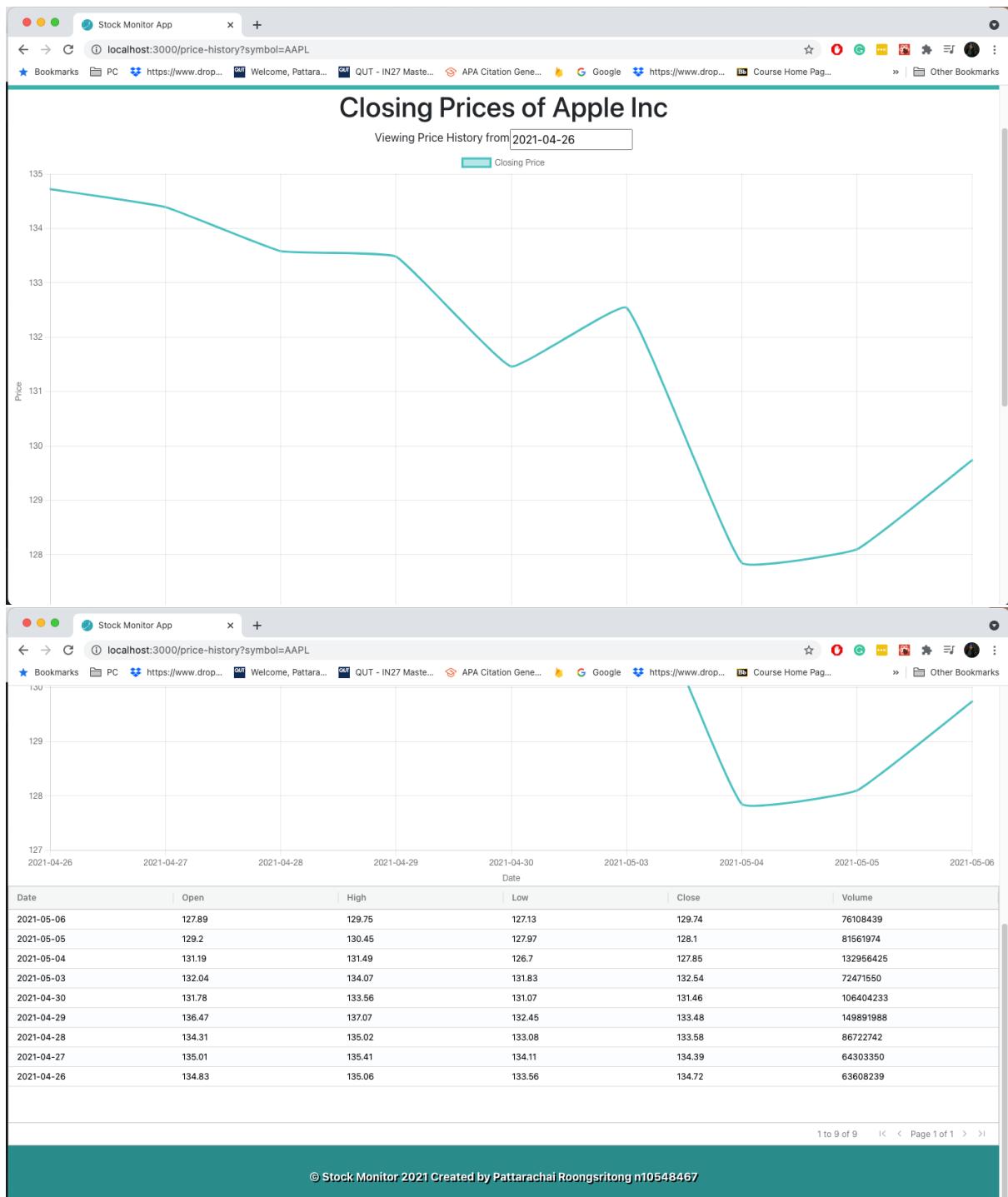


Figure 15. Select start date within 100 days

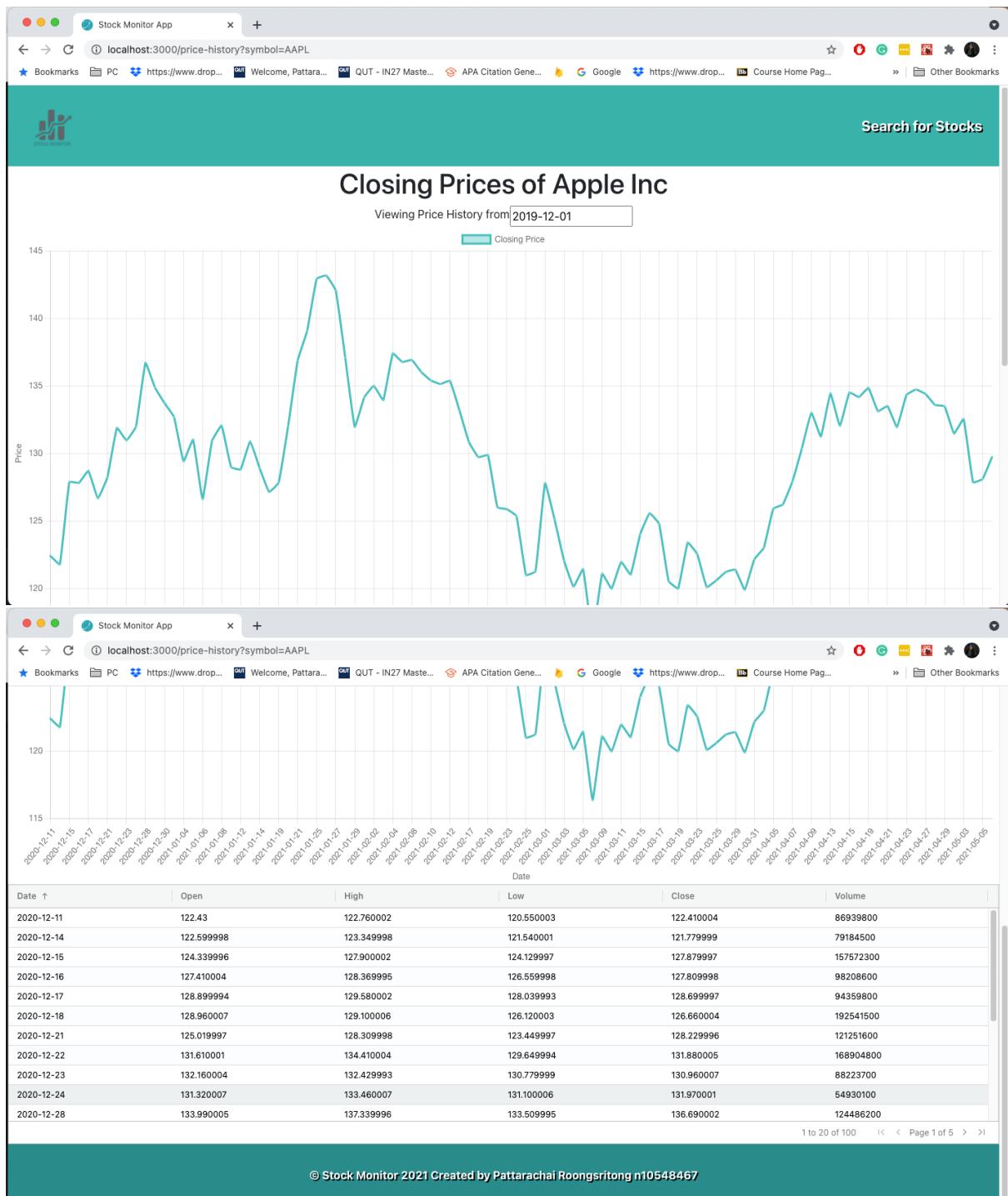


Figure 16. Select start date more than 100 days

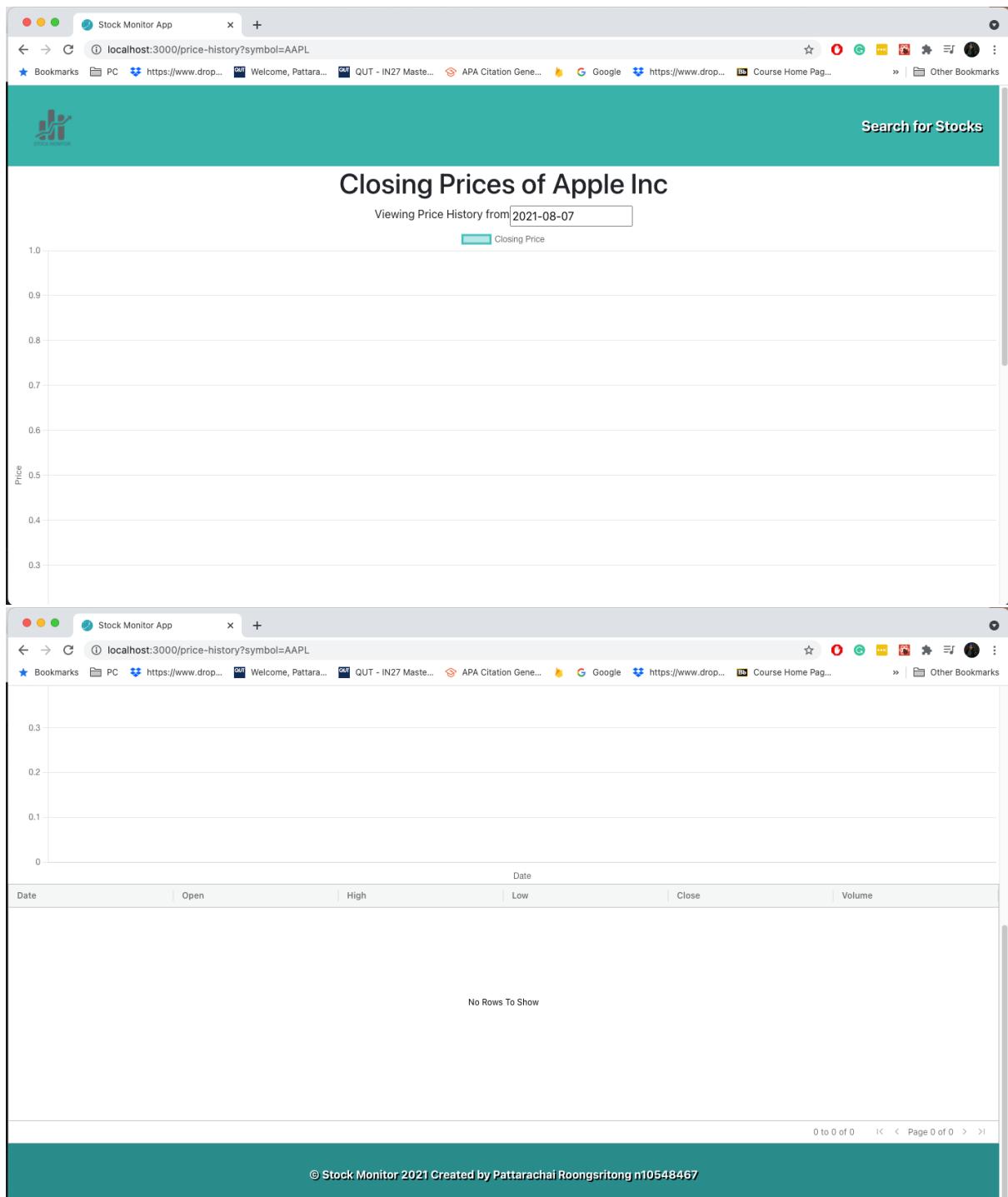


Figure 17. Select start date in the future

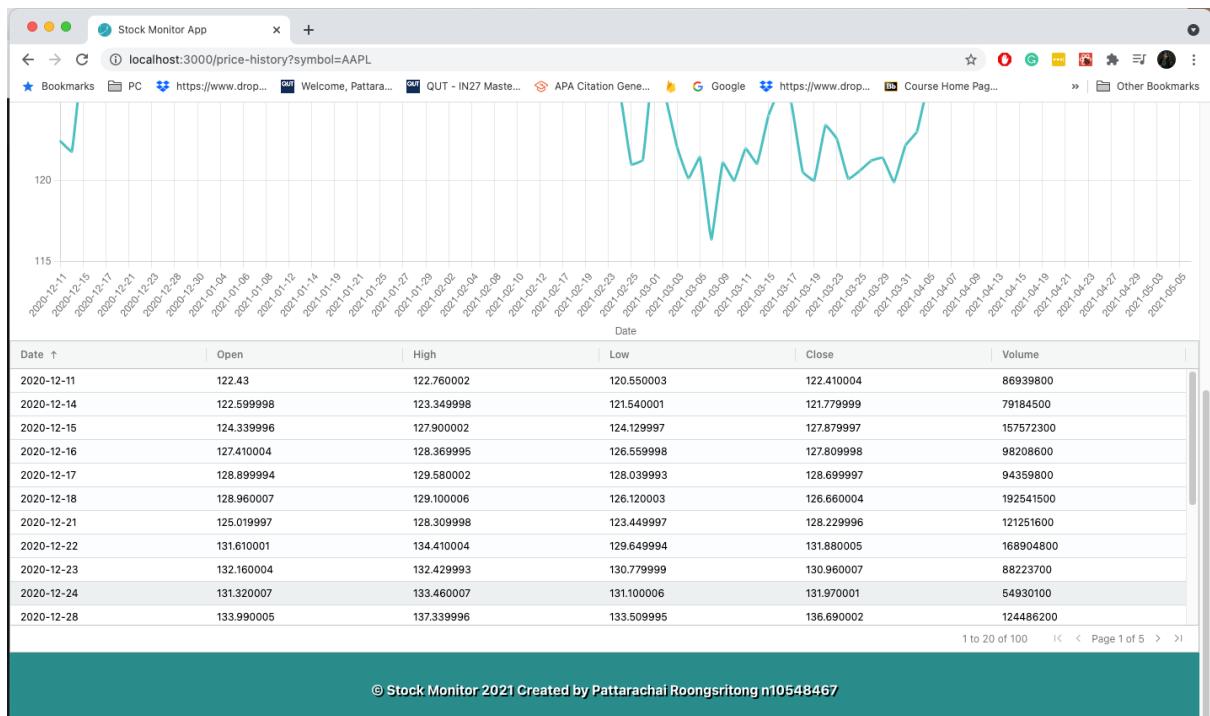


Figure 18. Sort a column