

FA25: DATA-200 Sec 72 - Comp Programming

LAB 2

Name: Siva Surya Chandran

ID: 019130215

The project will provide students with experience creating applications in Python. Students will use object-oriented techniques to develop a stock tracking application. The application will have both console and GUI (Graphical User Interfaces). By processing the historical stock data, profit/loss reports can be generated. An embedded database management system will allow users to save and retrieve stock data. The system will use the Python libraries to create charts and get historical stock data from web sites. Students will learn how to use the GitHub platform for their project.

In this part of the project, you will implement the ability to get stock price history from a web site, or by importing a .csv file.

ENVIRONMENT SETUP

As mentioned, I have downloaded and setup the environment for lab_2

Name	Date modified	Type	Size
.venv	04-12-2025 11:57	File folder	
__pycache__	09-12-2025 13:45	File folder	
fixtures	09-12-2025 13:51	File folder	
tools	09-12-2025 13:50	File folder	
requirements	07-12-2025 20:49	Text Document	1 KB
SJSU_Lab2_Stock_Analysis	04-12-2025 11:07	Chrome PDF Doc...	1,942 KB
stock_class	07-12-2025 22:57	Python Source File	7 KB
stock_console	07-12-2025 20:49	Python Source File	11 KB
stock_data	09-12-2025 13:45	Python Source File	7 KB
stock_GUI	07-12-2025 20:49	Python Source File	17 KB
stocks	09-12-2025 13:52	Data Base File	28 KB
stocks	04-12-2025 11:07	Python Source File	1 KB
utilities	07-12-2025 20:49	Python Source File	3 KB

I have added all the requirements (that need to be downloaded) in *requirements.txt* to download prior to the test execution

```
requirements.txt
1 matplotlib
2 beautifulsoup4
3 selenium
4 pandas
5 lxml
6 requests
```

This is how the vscode env is set

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a tree structure of files in the "LAB_2" folder, including stock_class.py, stock_console.py (the active file), stock_data.py, stock_GUI.py, stocks.db, stocks.py, and utilities.py.
- Code Editor:** Displays the content of stock_console.py. The code defines a main() function that imports stock_console and stock_GUI modules. It checks if the script is run as a script (if __name__ == "__main__") and calls the main() function of the stock_GUI module. A comment indicates that the program starts here.
- Terminal:** Shows the command PS C:\Users\Admin\Desktop\FALL25\200\Lab_2> & C:/Users/Admin/Desktop/FALL25/200/Lab_2/.venv/Scripts/python.exe c:/Users/Admin/Desktop/FALL25/200/Lab_2/stocks.py being run.

IMPLEMENTATION OF THE CONSOLE

Running the console version by uncommenting the stock_console.main()

Running stocks.py from console using the command typed in below.

The screenshot shows the VS Code interface with the following details:

- Code Editor:** Displays the content of stocks.py. The main() function is now uncommented, allowing the program to start via the console.
- Terminal:** Shows the command PS C:\Users\Admin\Desktop\FALL25\200\Lab_2> & C:/Users/Admin/Desktop/FALL25/200/Lab_2/.venv/Scripts/python.exe c:/Users/Admin/Desktop/FALL25/200/Lab_2/stocks.py being run.

Menu option opens up on running the stocks.py

The screenshot shows the VS Code interface with the following details:

- Code Editor:** Displays the content of stocks.py, specifically the main_menu() function which prints a menu for the Stock Analyzer.
- Terminal:** Shows the output of the program after running stocks.py. It displays the menu options and then asks for user input.

```
Stock Analyzer ---  
1 - Manage Stocks (Add, Update, Delete, List)  
2 - Add Daily Stock Data (Date, Price, Volume)  
3 - Show Report  
4 - Show Chart  
5 - Manage Data (Save, Load, Retrieve)  
0 - Exit Program  
Enter Menu Option: []
```

Entering 1 - Selecting Manage stocks from Menu Option to further test functionalities.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Manage Stocks ---
1 - Add Stock
2 - Update Shares
3 - Delete Stock
4 - List Stocks
0 - Exit Manage Stocks
Enter Menu Option: 
```

Entering 1 - Testing the Adding stocks functionality by adding Ticker symbol, Company name and the number of shares.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Add Stock ---
Enter Stock Symbol (or 0 to cancel): NVID
Enter Company Name: NVIDIA
Enter number of shares: 256
Added NVID.
Press Enter to continue... 
```

Entering 4 - Testing the List stocks functionality to check whether the added stock is reflected

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Manage Stocks ---
1 - Add Stock
2 - Update Shares
3 - Delete Stock
4 - List Stocks
0 - Exit Manage Stocks
Enter Menu Option: 
```



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Tracked Stocks ---
NVID - NVIDIA - 256.0 shares
Press Enter to continue... 
```

Entering 2 - Testing the Update Share functionality

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Update Shares ---
1 - Buy Shares
2 - Sell Shares
0 - Return
Enter Option: 
```

Entering 1 in Update Shares - Testing the Buy shares by selecting the ticker symbol and the number of shares

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Buy Shares ---
Stock List: [NVID]
Enter symbol to buy (or 0 to cancel): NVID
Enter shares to buy: 3698
Bought 3698.0 shares of NVID.
Press Enter to continue... 
```

Entering 2 in Update Shares - Testing the Sell shares by selecting the ticker symbol and the number of shares

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ⌂

Sell Shares ---
Stock List: [NVID]
Enter symbol to sell (or 0 to cancel): NVID
Enter shares to sell: 2563
Sold 2563.0 shares of NVID.
Press Enter to continue... 
```

Returning to Manage Stocks menu

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | ⌂

Manage Stocks ---
1 - Add Stock
2 - Update Shares
3 - Delete Stock
4 - List Stocks
0 - Exit Manage Stocks
Enter Menu Option: []
```

Entering 4 - Testing the List Stocks - Listing the stocks that I added. It displays the ticker symbol, company name and number of shares.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | ⌂

Tracked Stocks ---
NVID - NVIDIA - 1391.0 shares
RAV - RAVAN - 25863.0 shares
Press Enter to continue...[]
```

Entering 3 - Testing deleting share functionality, adding a checker as delete has to handled carefully.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | ⌂

Delete Stock ---
Enter symbol to delete (or 0 to cancel): RAV
Deleted RAV.
Press Enter to continue...[]
```

Entering 4 - Testing the List Stocks - Listing the stocks that display stocks without the stocks I deleted. It displays the ticker symbol, company name and number of shares.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | ⌂

Tracked Stocks ---
NVID - NVIDIA - 1391.0 shares
Press Enter to continue...[]
```

Returning to main menu

```
Stock Analyzer ---
1 - Manage Stocks (Add, Update, Delete, List)
2 - Add Daily Stock Data (Date, Price, Volume)
3 - Show Report
4 - Show Chart
5 - Manage Data (Save, Load, Retrieve)
0 - Exit Program
Enter Menu Option: []
```

Entering 2 - Testing the Add daily stock data functionality by selecting the ticker symbol, Date, Closing price and the number of shares

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | ⌂

Add Daily Stock Data ---
Enter symbol (or 0 to cancel): NVID
Enter Date (m/d/yy): 10/05/25
Enter closing price: 226
Enter volume: 146
Daily data added.
Press Enter to continue...[]
```

Returning to main menu

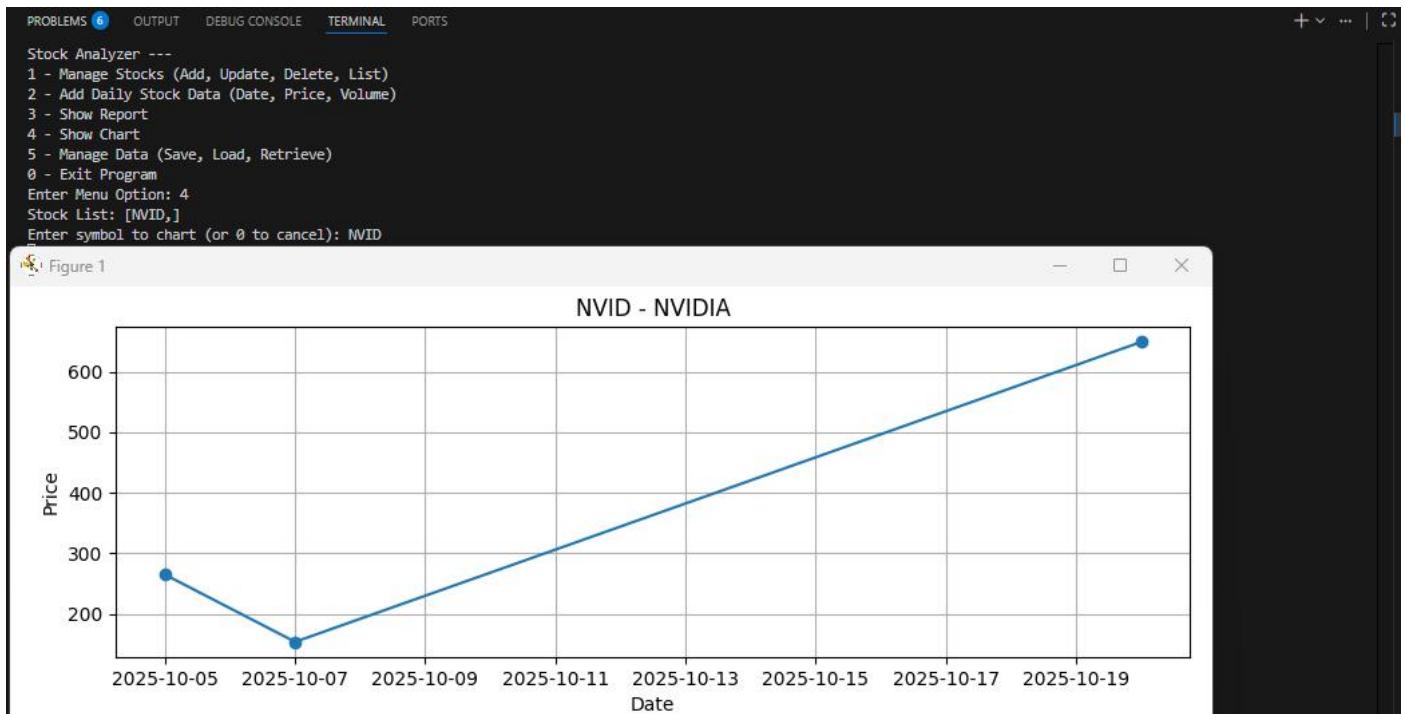
```
Stock Analyzer ---
1 - Manage Stocks (Add, Update, Delete, List)
2 - Add Daily Stock Data (Date, Price, Volume)
3 - Show Report
4 - Show Chart
5 - Manage Data (Save, Load, Retrieve)
0 - Exit Program
Enter Menu Option: []
```

Entering 3 - Testing Stock Report - Displaying the report for each stock that was entered earlier

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂

Stock Report ---
NVID - NVIDIA - 1391.0 shares - 2 data points
Last: 11/06/25 - $645.00 - Vol: 1485.0
Press Enter to continue...[]
```

Entering 4 - Testing the Show Chart - Displaying chart for the selected stock from the list



Entering 5 - To explore the Manage data menu option

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂

Manage Data ---
1 - Save Data
2 - Load Data
3 - Retrieve From Web
4 - Import CSV
0 - Return
```

Entering 1 - Testing Save Data - Saving current data to database

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂

Manage Data ---
1 - Save Data
2 - Load Data
3 - Retrieve From Web
4 - Import CSV
0 - Return
Enter Option: 1
Data saved.
Press Enter to continue...[]
```

Entering 2 - Testing Load Data - Loading the data that was saved earlier

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂

Manage Data ---
1 - Save Data
2 - Load Data
3 - Retrieve From Web
4 - Import CSV
0 - Return
Enter Option: 2
Data loaded.
Press Enter to continue...[]
```

Entering 3 - Testing the important feature - Retrieving data from web which is yahoo! Finance.

By entering the Start date and End date within which we have to retrive the data

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂
Retrieve From Web ---
Enter start date (m/d/yy): 10/05/25
Enter end date (m/d/yy): 12/05/25
Retrieved 129 records.
Press Enter to continue...[]
```

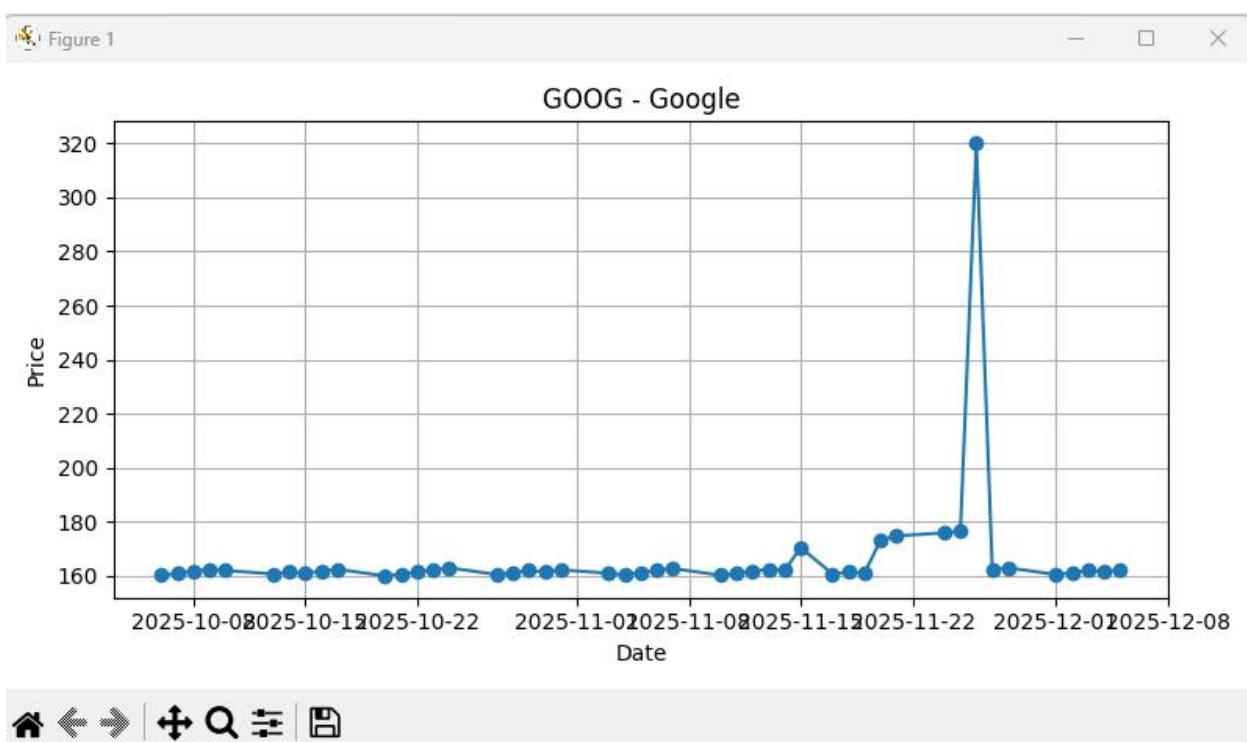
Entering 1 - Testing Save Data - Saving current data to database (after retrieval)

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂
Manage Data ---
1 - Save Data
2 - Load Data
3 - Retrieve From Web
4 - Import CSV
0 - Return
Enter Option: 1
Data saved.
Press Enter to continue...[]
```

Entering 2 - Testing Load Data - Loading the data that was saved now (after retrieval)

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ⌂ | ⌂
Manage Data ---
1 - Save Data
2 - Load Data
3 - Retrieve From Web
4 - Import CSV
0 - Return
Enter Option: 2
Data loaded.
Press Enter to continue...[]
```

Entering back to Main menu and Entering 4 - to check the chart of GOOG - after retrieval



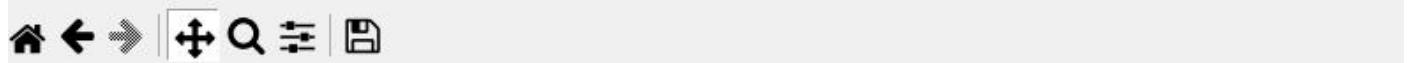
Again Entering Manage data menu

Entering 4 - Testing the feature of Importing from CSV file.

(Using NASDAQ as the csv files were paid in yahoo! Finance.)

```
Import Stock History From CSV File ---  
This will import data for a single stock from a Yahoo! Finance CSV file.  
Enter Ticker Symbol: AMZN  
Enter CSV Filename (with full path): AMAZON COM INC (01-03-2022 - 12-01-2025).csv  
Import Complete for AMZN.  
Press Enter to Continue
```

Entering back to Main menu and Entering 4 - to check the chart of AMZN - To verify after importing,



END OF CONSOLE IMPLEMENTATION

Exiting the console by Entering 0

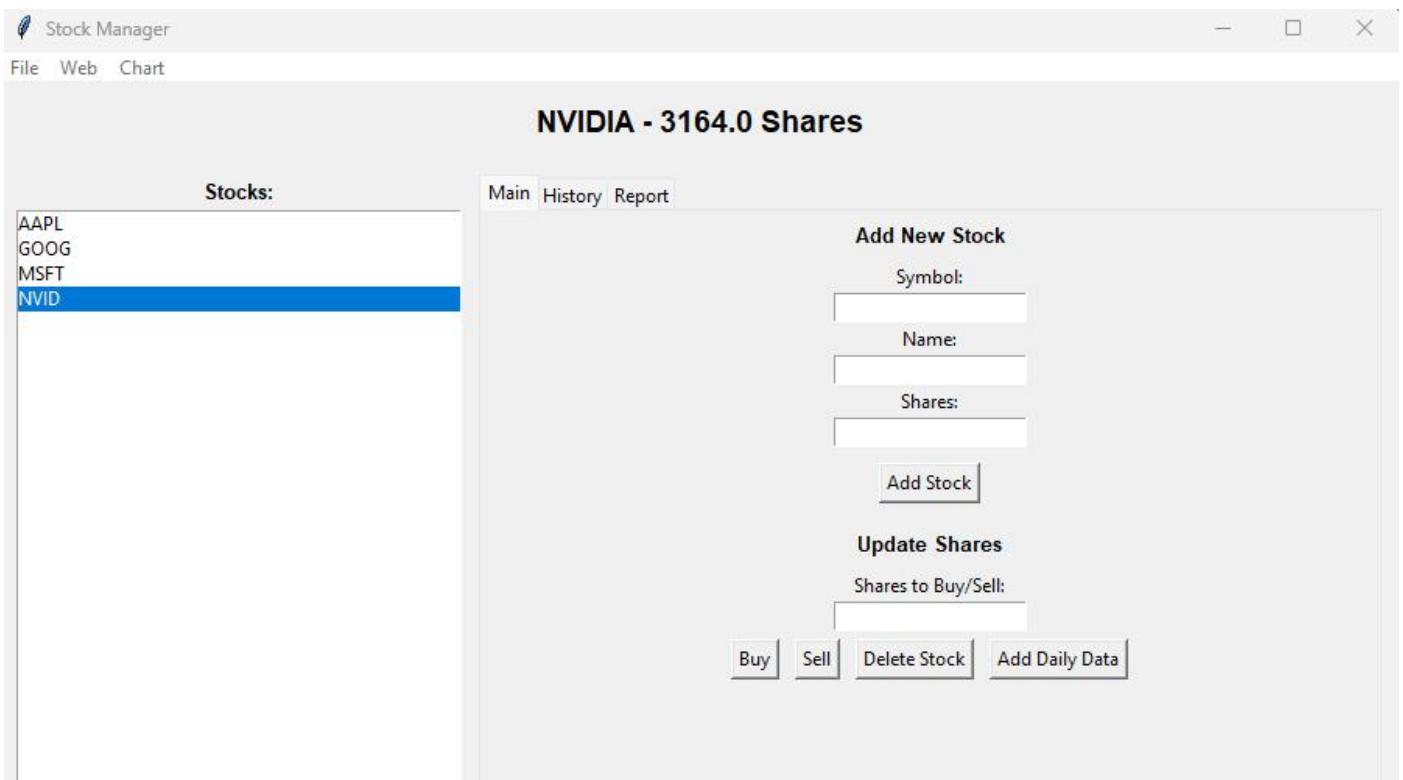
The screenshot shows a Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure under "LAB_2". The file "stock_console.py" is currently selected and has a red status bar icon indicating 6 errors.
- Editor View:** Displays the code for "stock_console.py". The main menu loop is shown, including options for charts, data management, and exiting the program.
- Bottom Status Bar:** Shows "PROBLEMS 6", "OUTPUT", "DEBUG CONSOLE", "TERMINAL" (which is active), and "PORTS".
- Terminal View:** Shows the command "PS C:\Users\Admin\Desktop\FALL25\200\Lab_2> []" at the bottom.

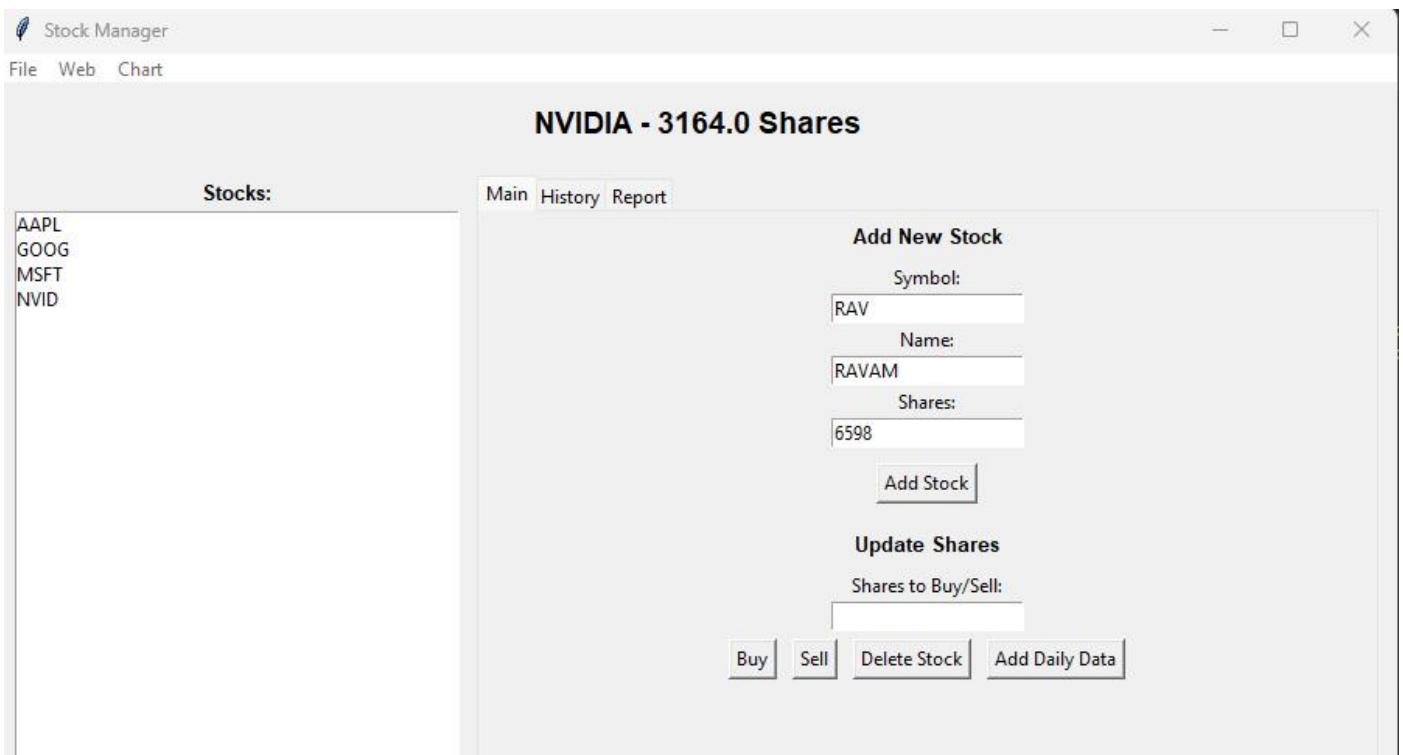
By processing the historical stock data, profit/loss reports can be generated. An embedded database management system will allow users to save and retrieve stock data. The system will use the Python libraries to create charts and get historical stock data from web sites

GUI (Graphical User Interfaces) IMPLEMENTATION

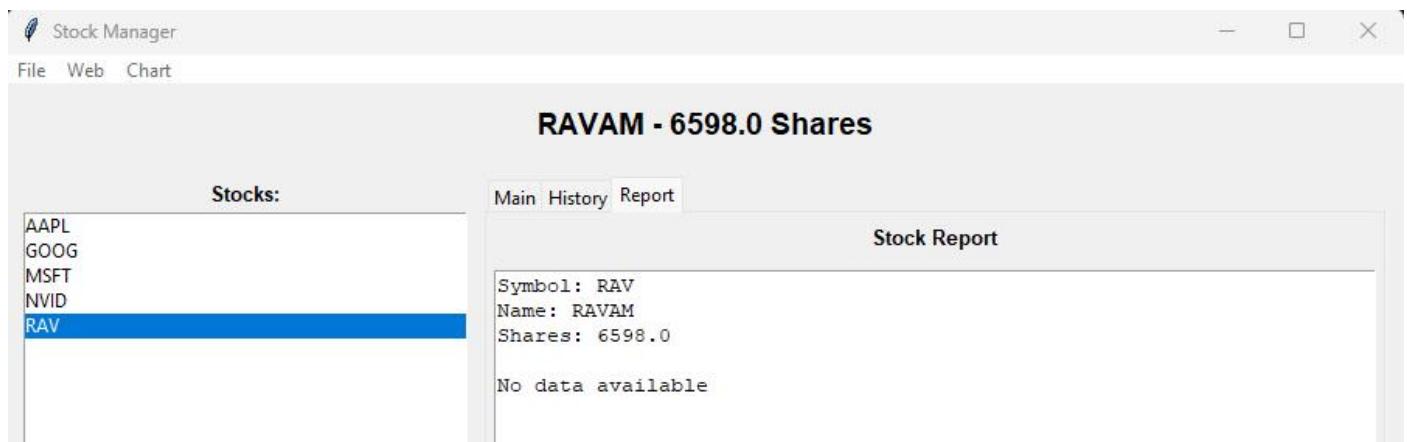
GUI Main menu screen



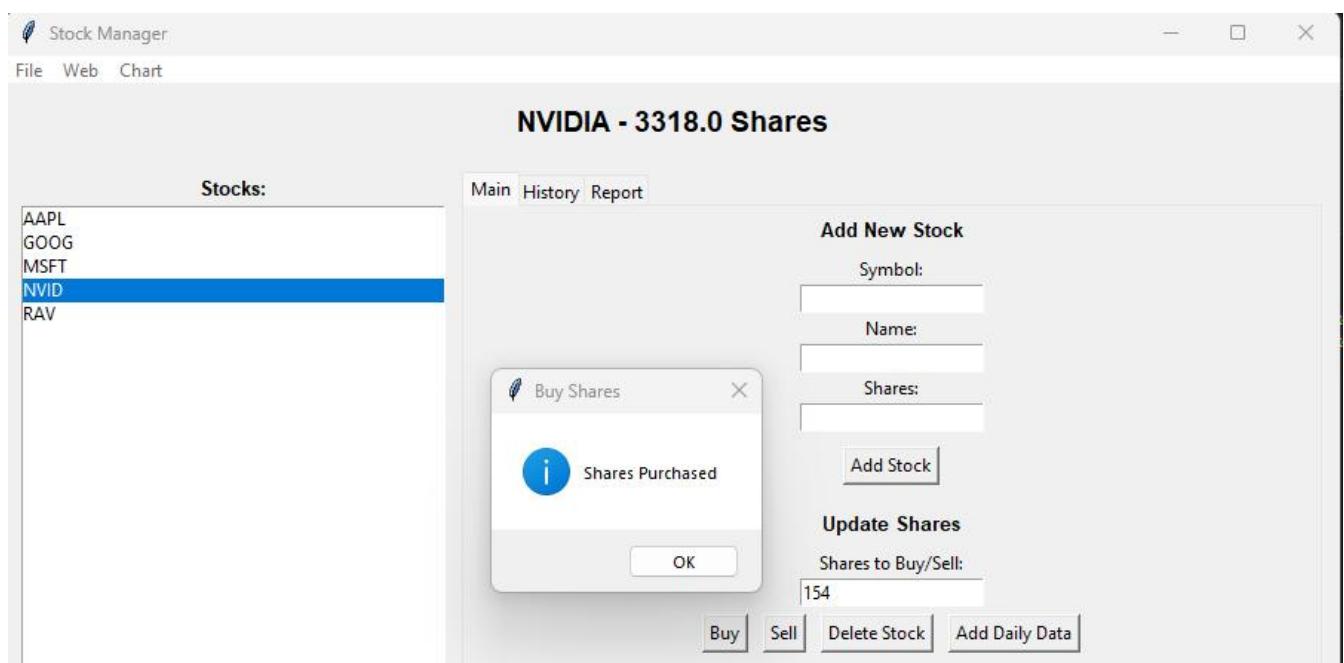
Testing Add New Stock feature by adding RAV



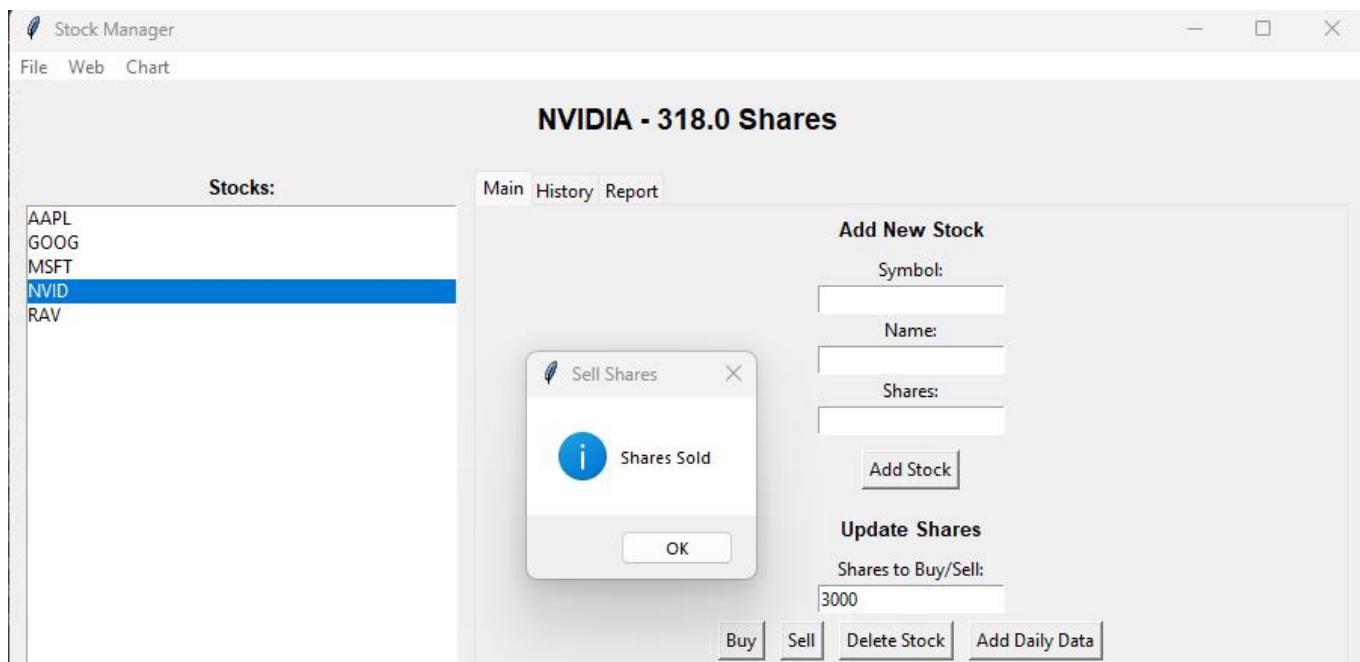
Testing whether the report is updated for the Stock



Testing Purchase stock by selecting stock - Entering shares count and press Buy



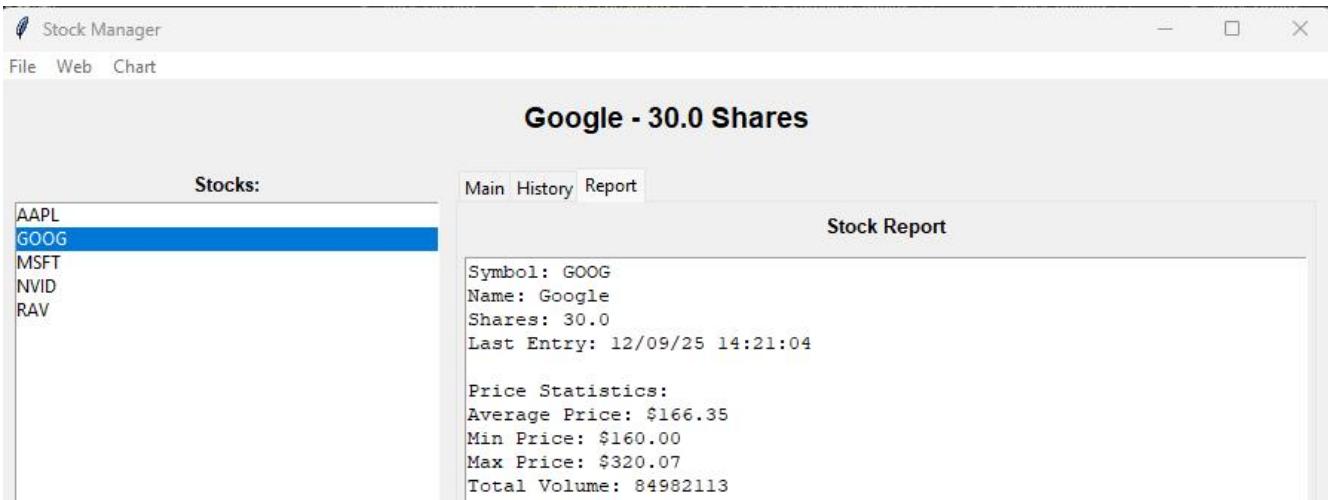
Testing Selling stock by selecting stock - Entering shares count and press Sell



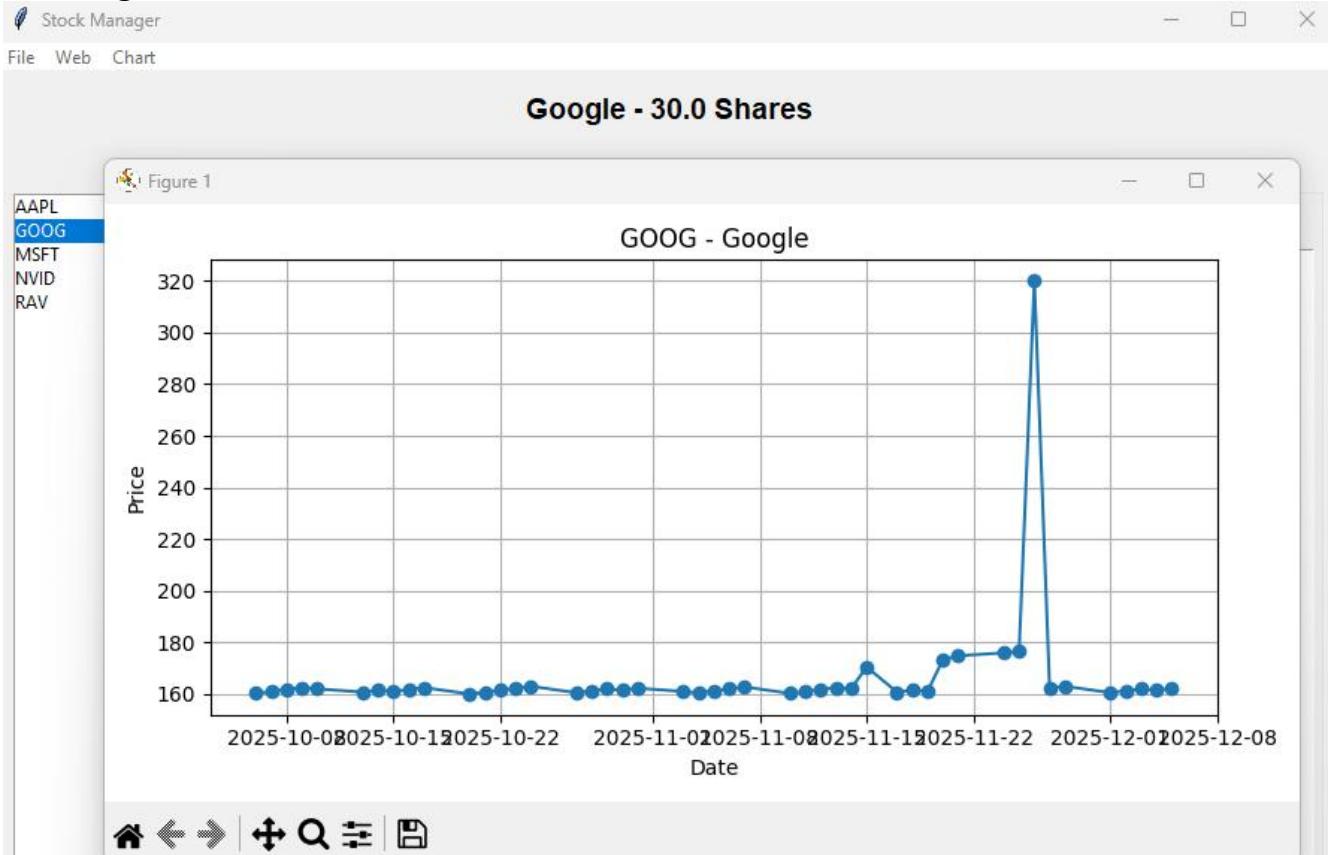
Checking whether the Daily stock data history is updated properly



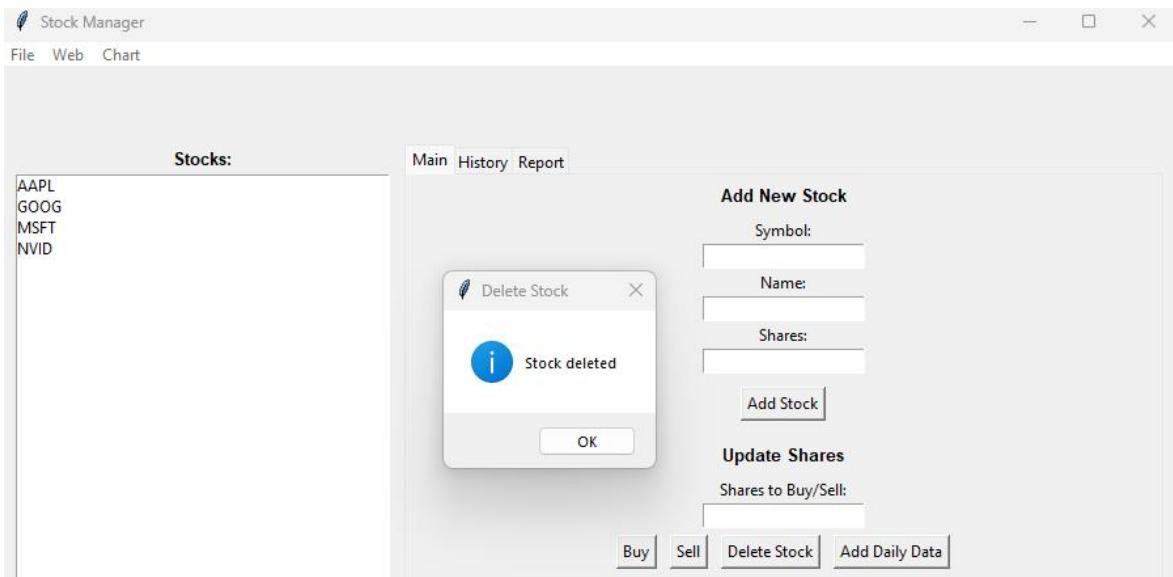
Checking whether the Share Report is updated properly



Checking the Chart of the Stock data



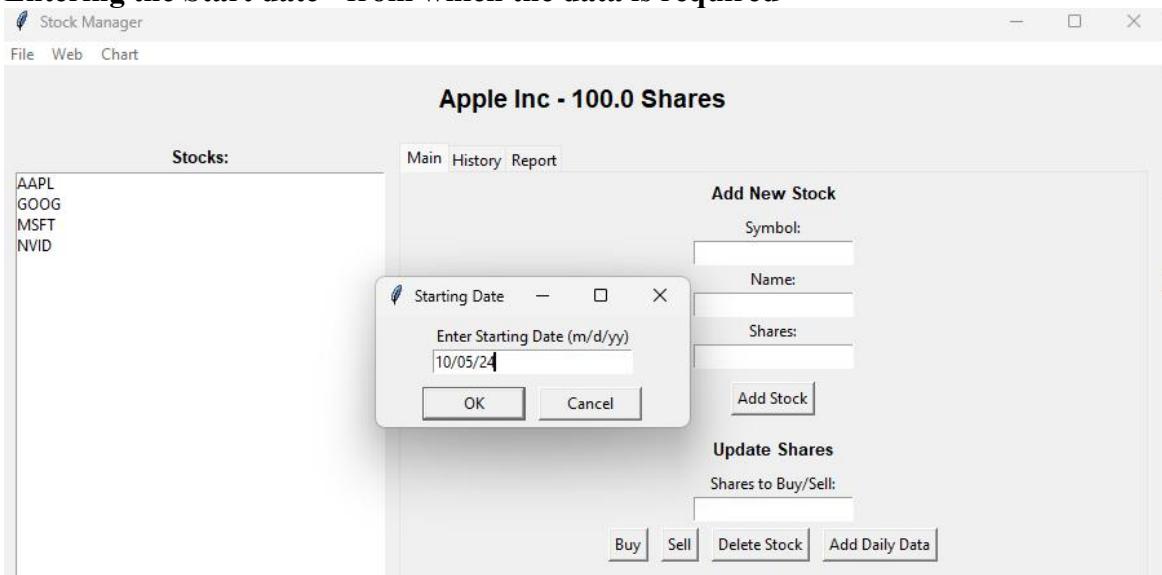
Checking the Delete Stock - by Deleting the stock RAV



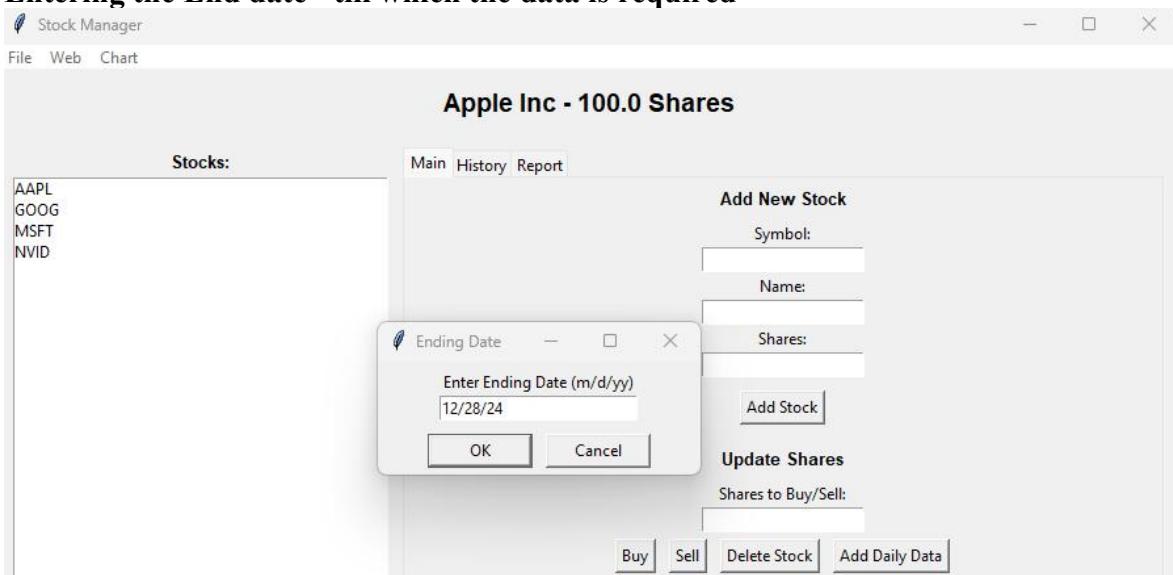
Retrieving Stock from web

Select Web from the left top and select Retrive from web option

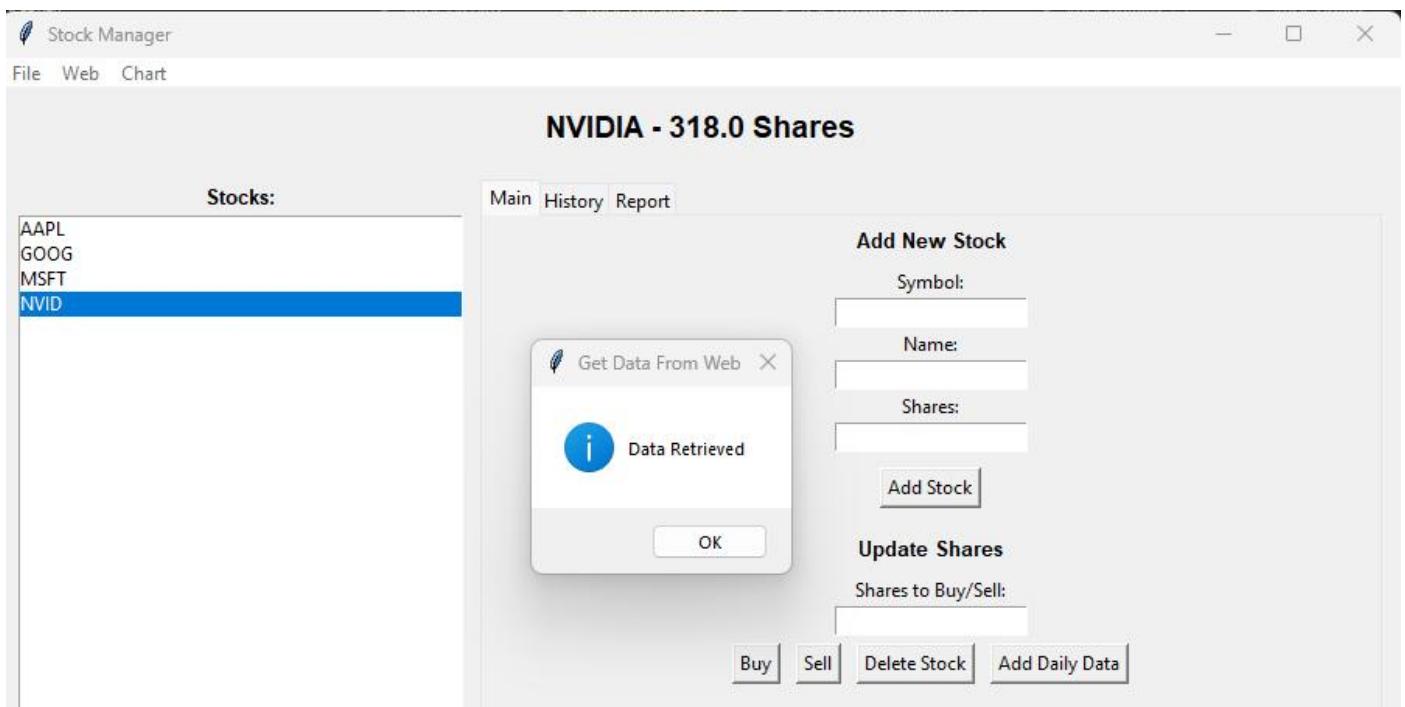
Entering the Start date - from which the data is required



Entering the End date - till which the data is required



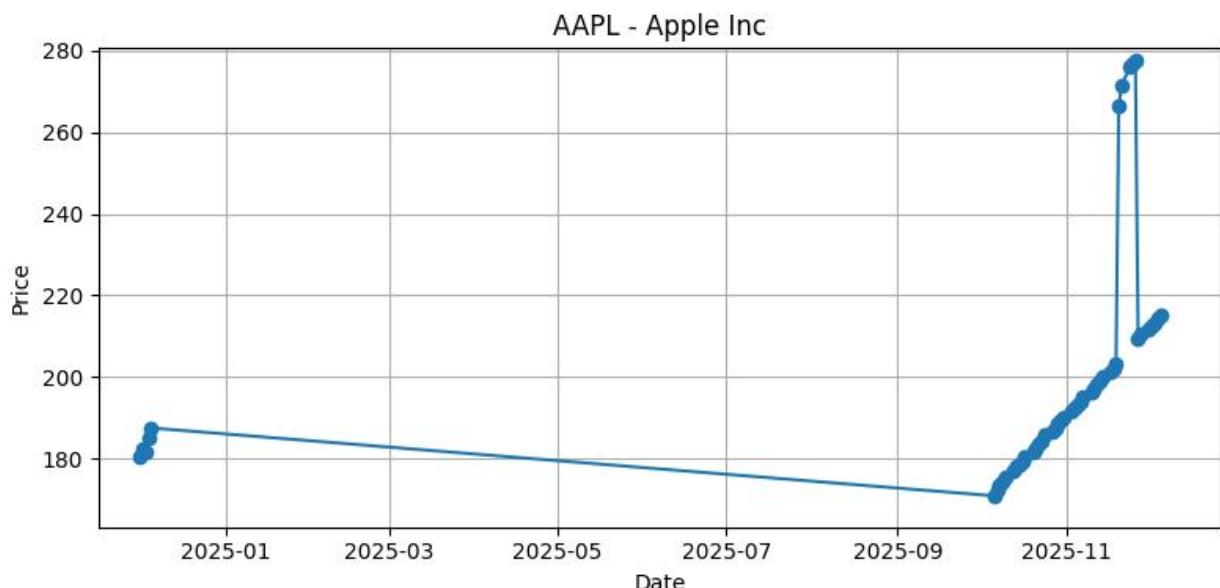
Now thw Data is retrieved from the Web



Checking whether the History shows the updated data between the date's entered



Checking whether the Chart is also updated



Importing data of a stock using csv file

File Web Chart

Amazon - 150.0 Shares

Tracked Stocks

AMZN

History Report

Date	Price	Volume
12/01/25	\$229.53	2029
11/24/25	\$233.22	1524
11/17/25	\$220.69	2976
11/10/25	\$234.69	1715
11/03/25	\$244.41	2805
10/27/25	\$244.22	4059
10/20/25	\$224.21	2039
10/13/25	\$213.04	2177
10/06/25	\$216.37	2403

✓ import Complete X

i AMZN Import Complete

OK

Validating the Chart



Github

All the Files of this lab_2 is uploaded to the git

Github link - https://github.com/sivasuryachandran/DATA_200_lab_2