FA25: DATA-200 Sec 72 - Comp Programming

LAB 2

Name: Siva Surya Chandran

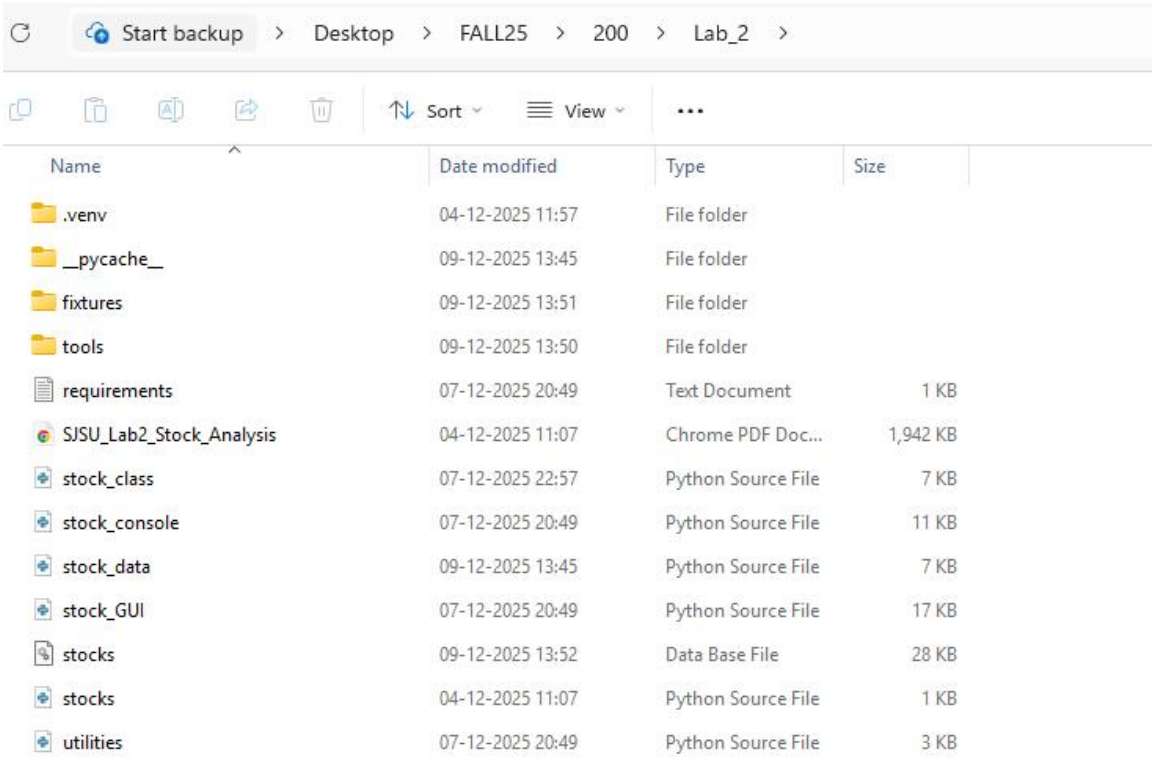
ID: 019130215

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.

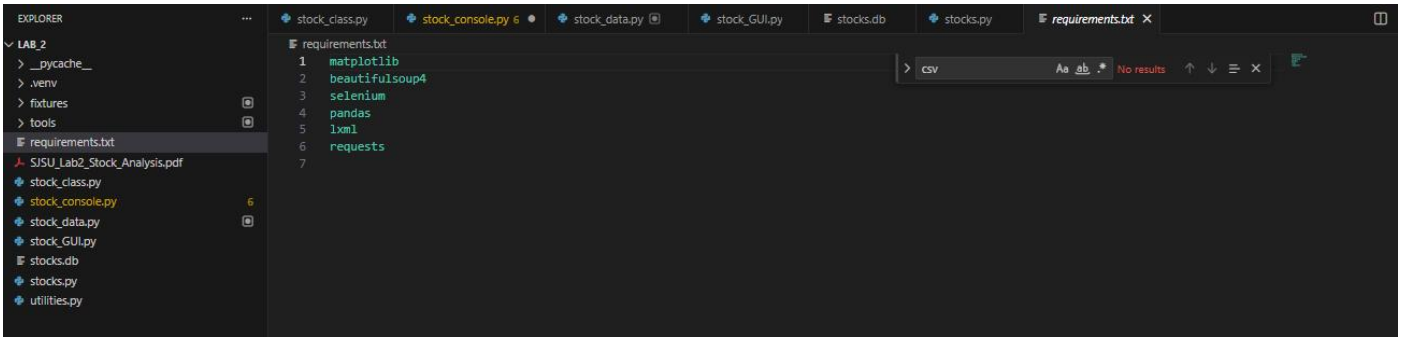
In this part of the project, you will implement the ability to get stock price history from a web site, or by importing a .csv file.

ENVIRONMENT SETUP

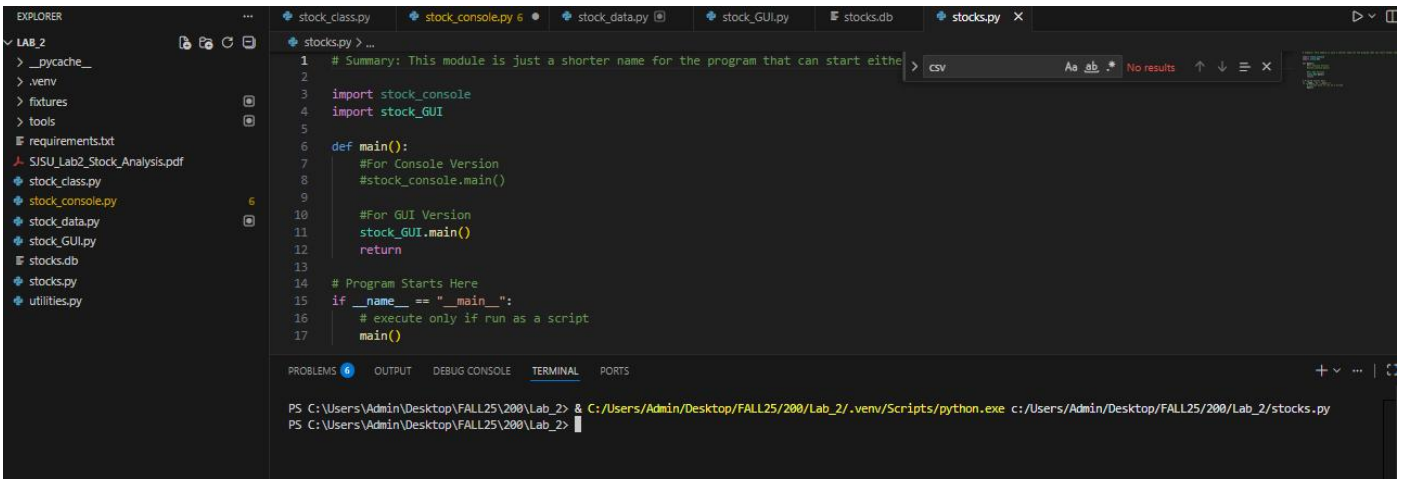
As mentioned, I have downloaded and setup the environment for lab_2



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



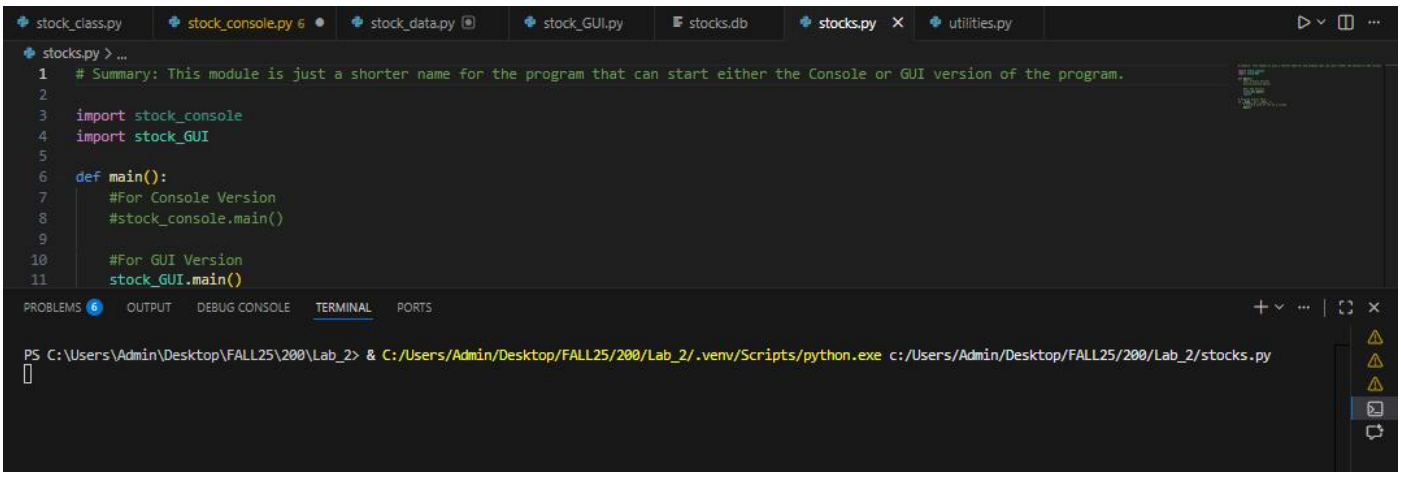
This is how the vscode env is set



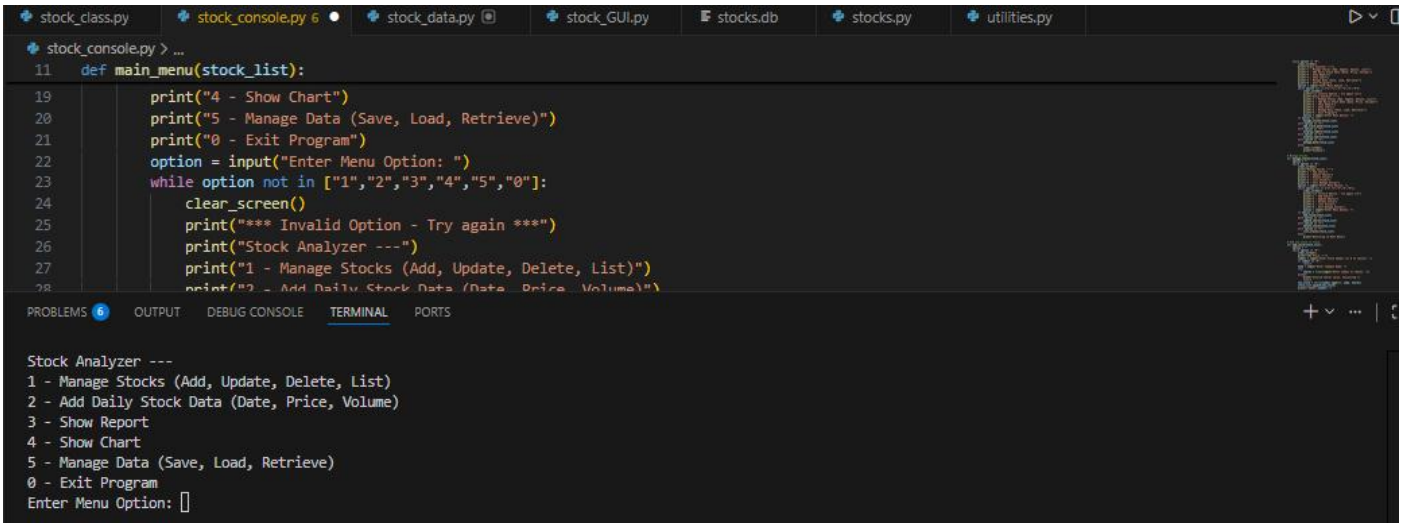
IMPLEMENTATION OF THE CONSOLE

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

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



Menu option opens up on running the stocks.py



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

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

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

Entering 2 - Testing the Update Share functionality

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

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

Returning to Manage Stocks menu

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

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

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.

Returning to main menu

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

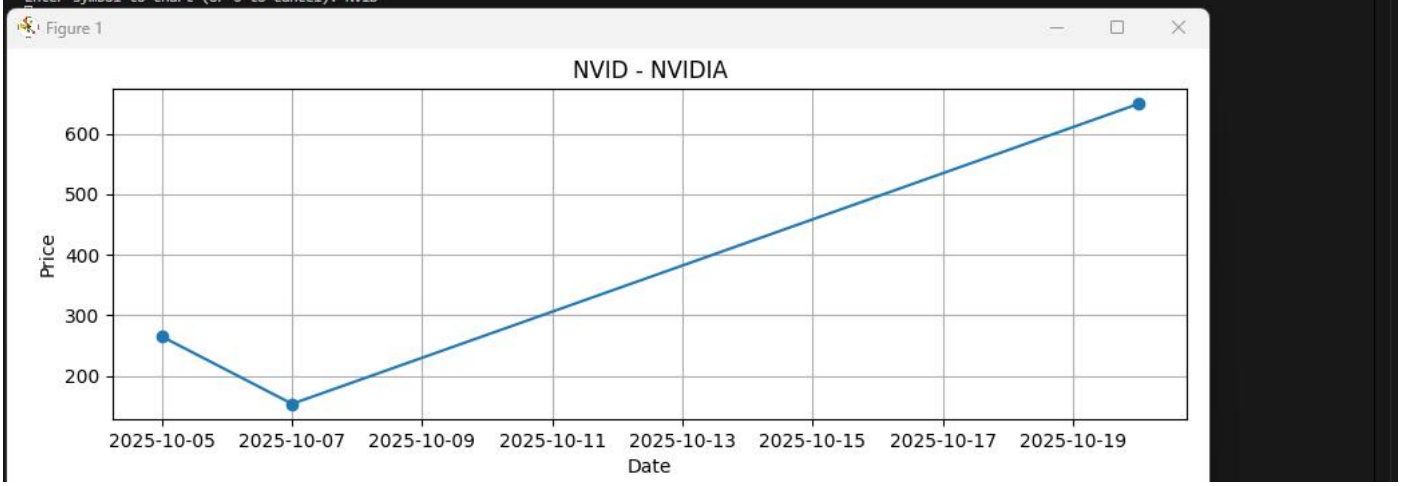
Returning to main menu

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

```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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: 4
Stock List: [NVID,]
Enter symbol to chart (or 0 to cancel): NVID
```



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...
```

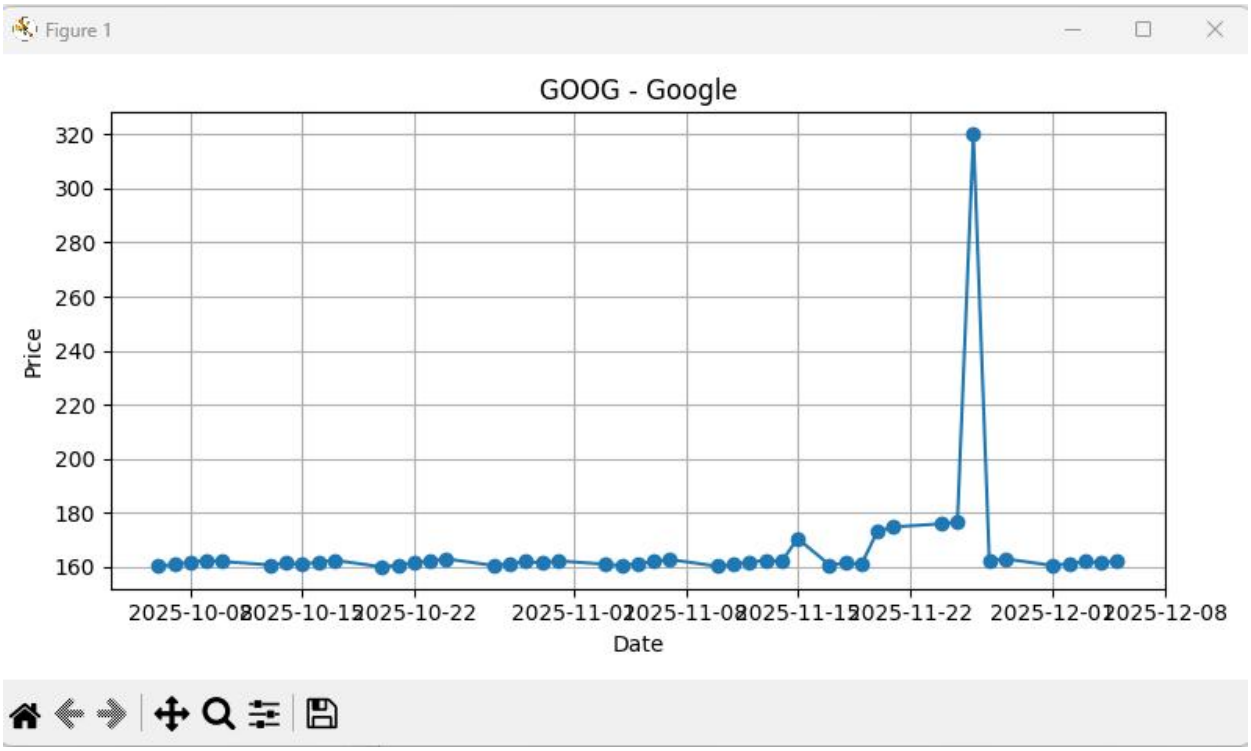
Entering 1 - Testing Save Data - Saving current data to database (after retrival)

```
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 retrival)

```
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 retrival



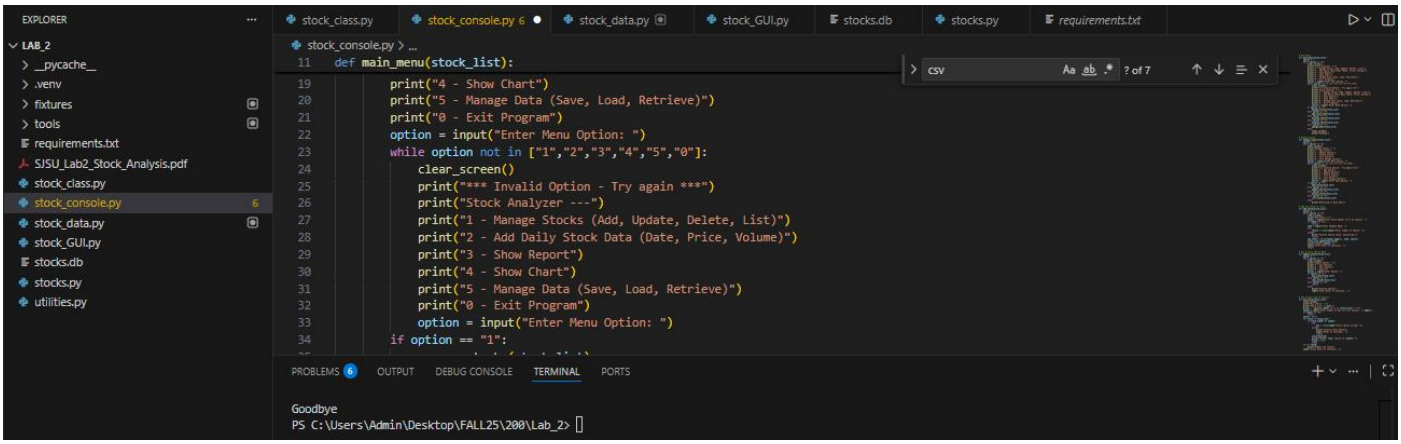
Again Entering Manage data menu
Entering 4 - Testing the feature of Importing from CSV file.
(Using NASDAQ as the csv files were paid in yahoo! Finance.)

```
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

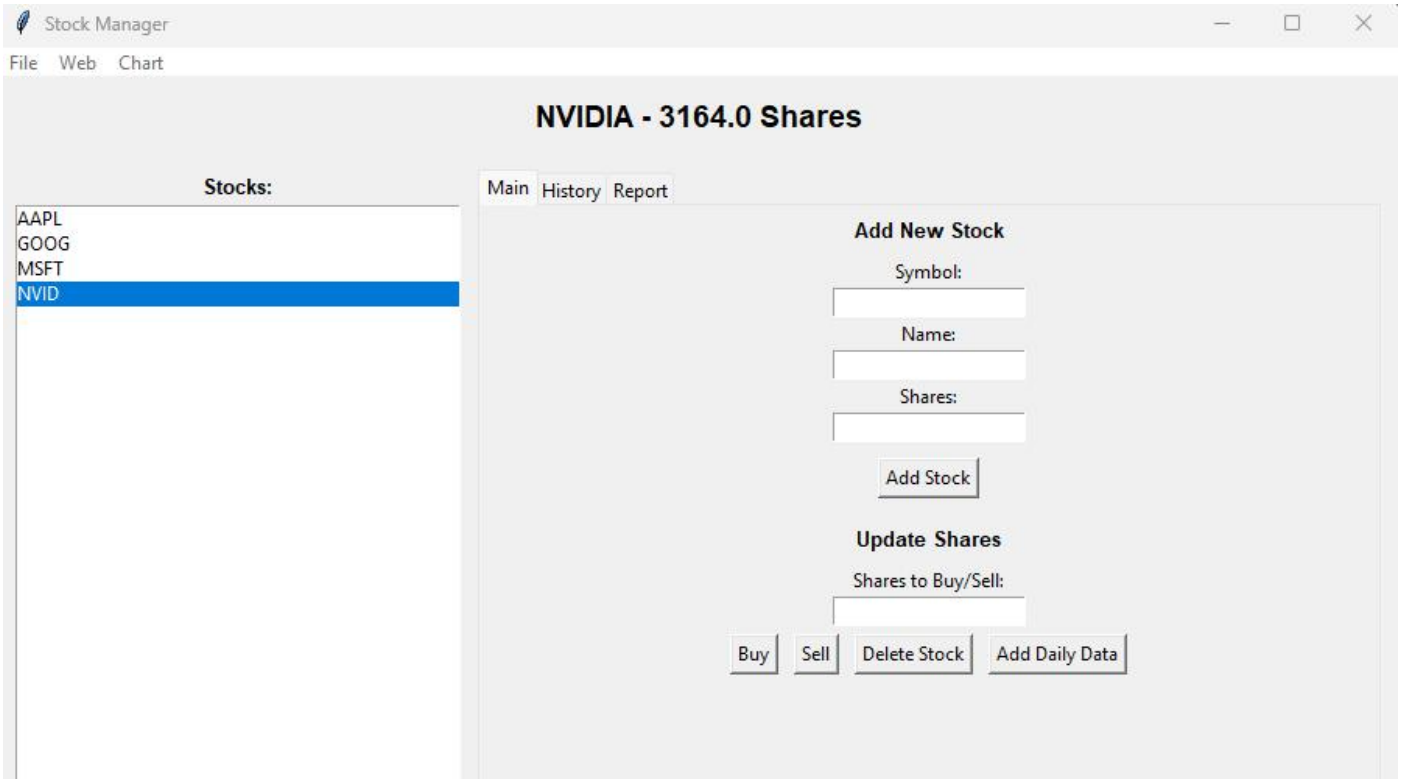
Exiting the console by Entering 0



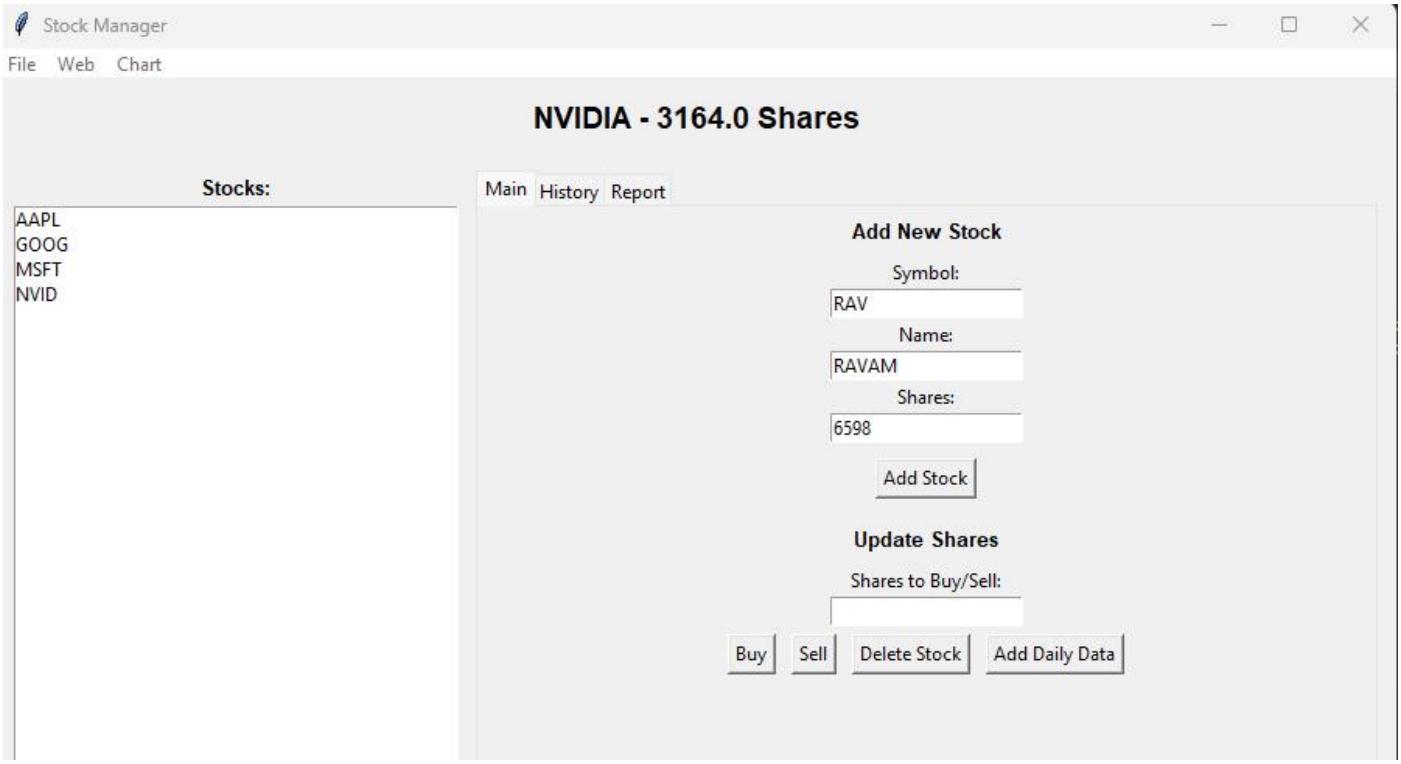
By processing the historical stock data, profit/loss reports can be generated. An embedded database management system will allow users to save and retrieve stock data. The system will use the Python libraries to create charts and get historical stock data from web sites

GUI (Graphical User Interfaces) IMPLEMENTATION

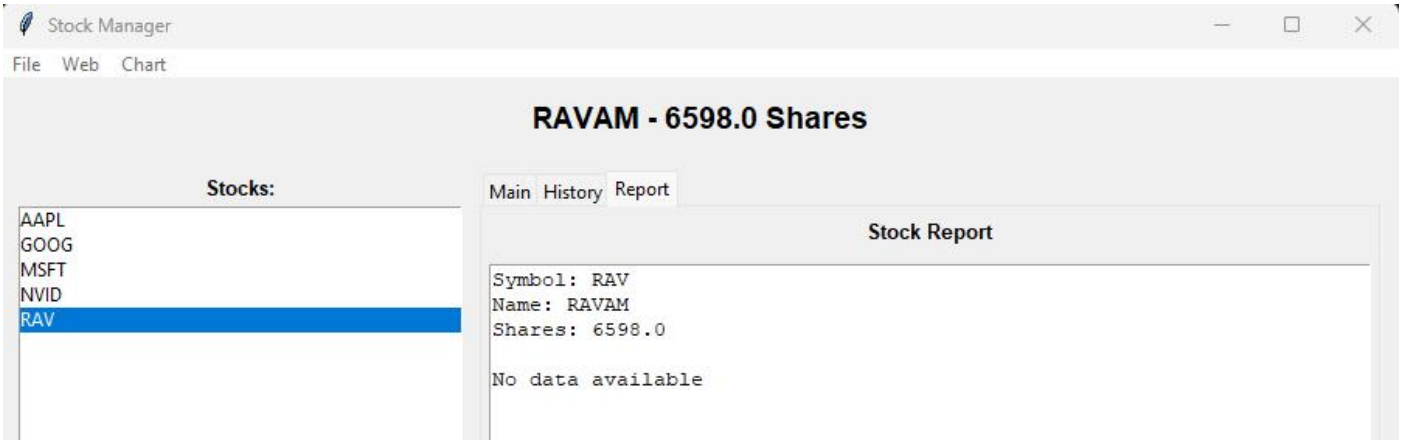
GUI Main menu screen



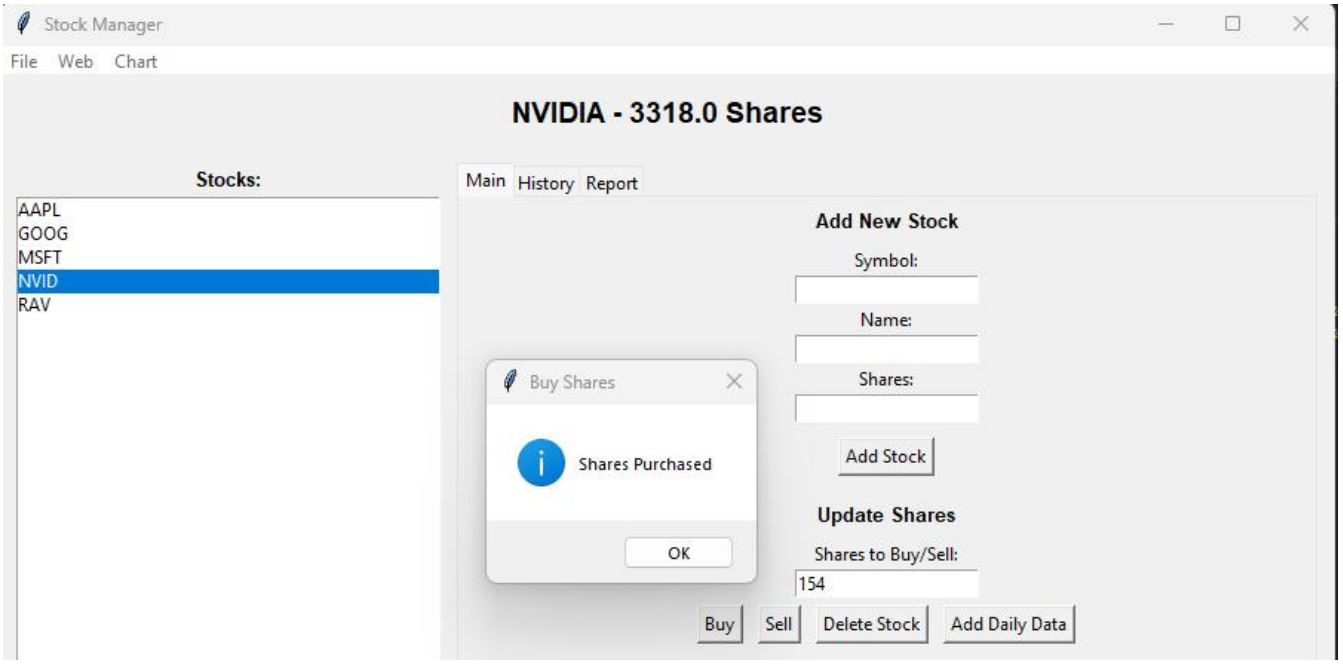
Testing Add New Stock feature by adding RAV



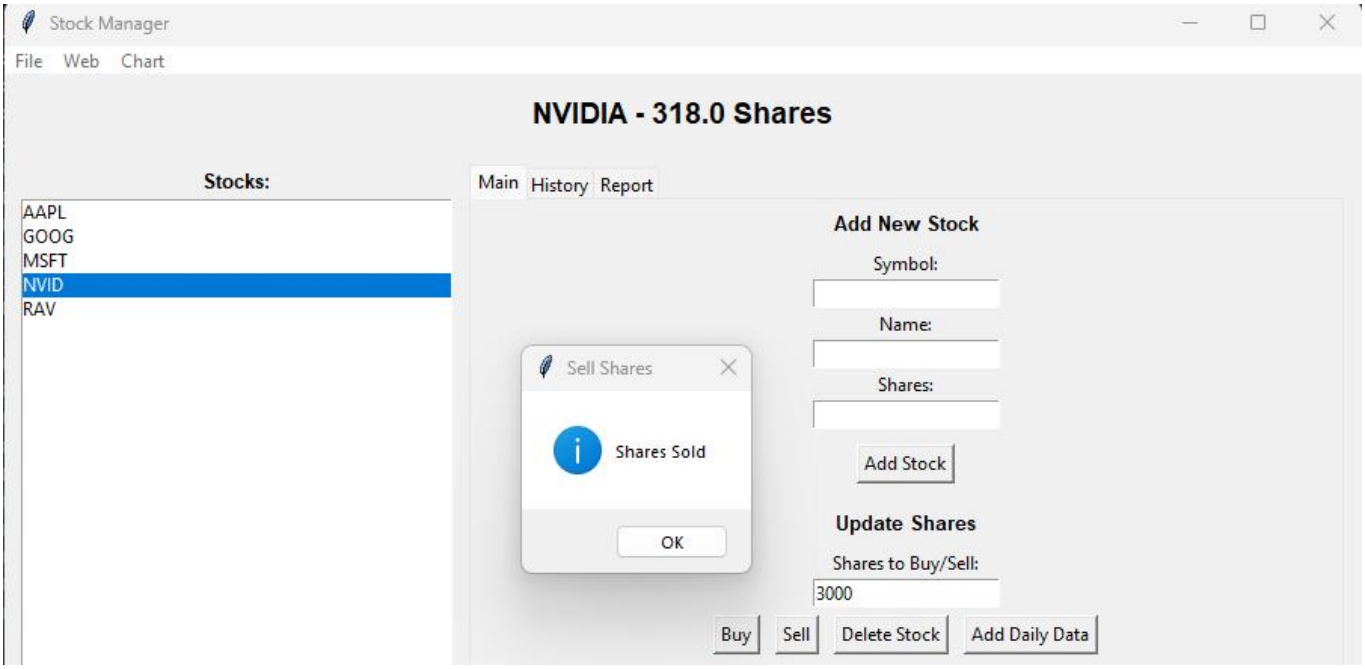
Testing whether the reoprt is updated for the Stock



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

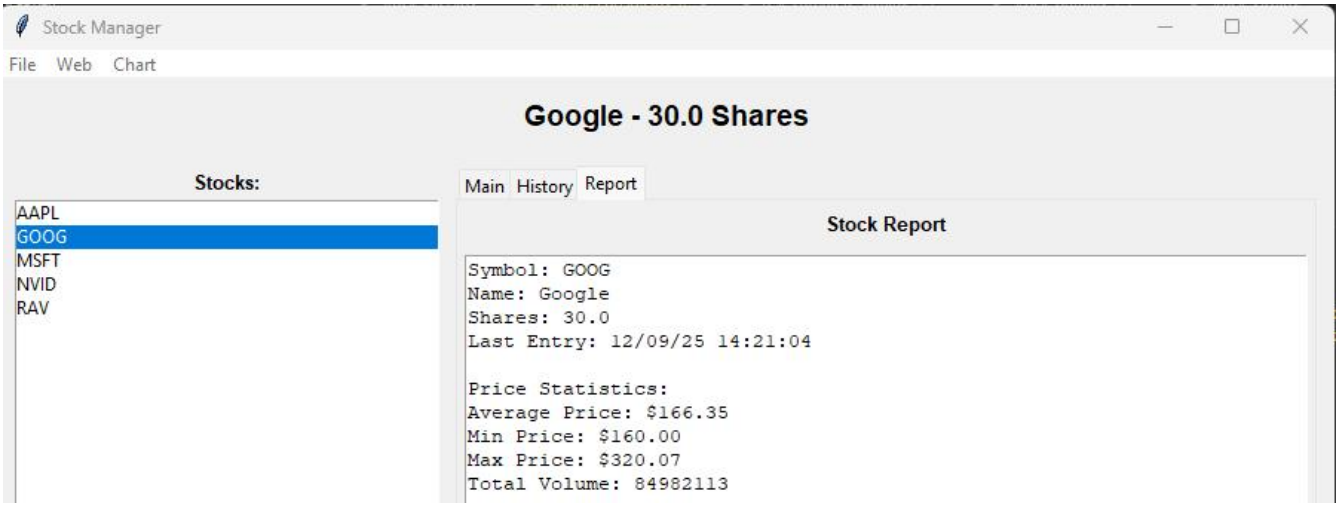


Testing Selling stock by selecting stock - Entering shares count and press Sell

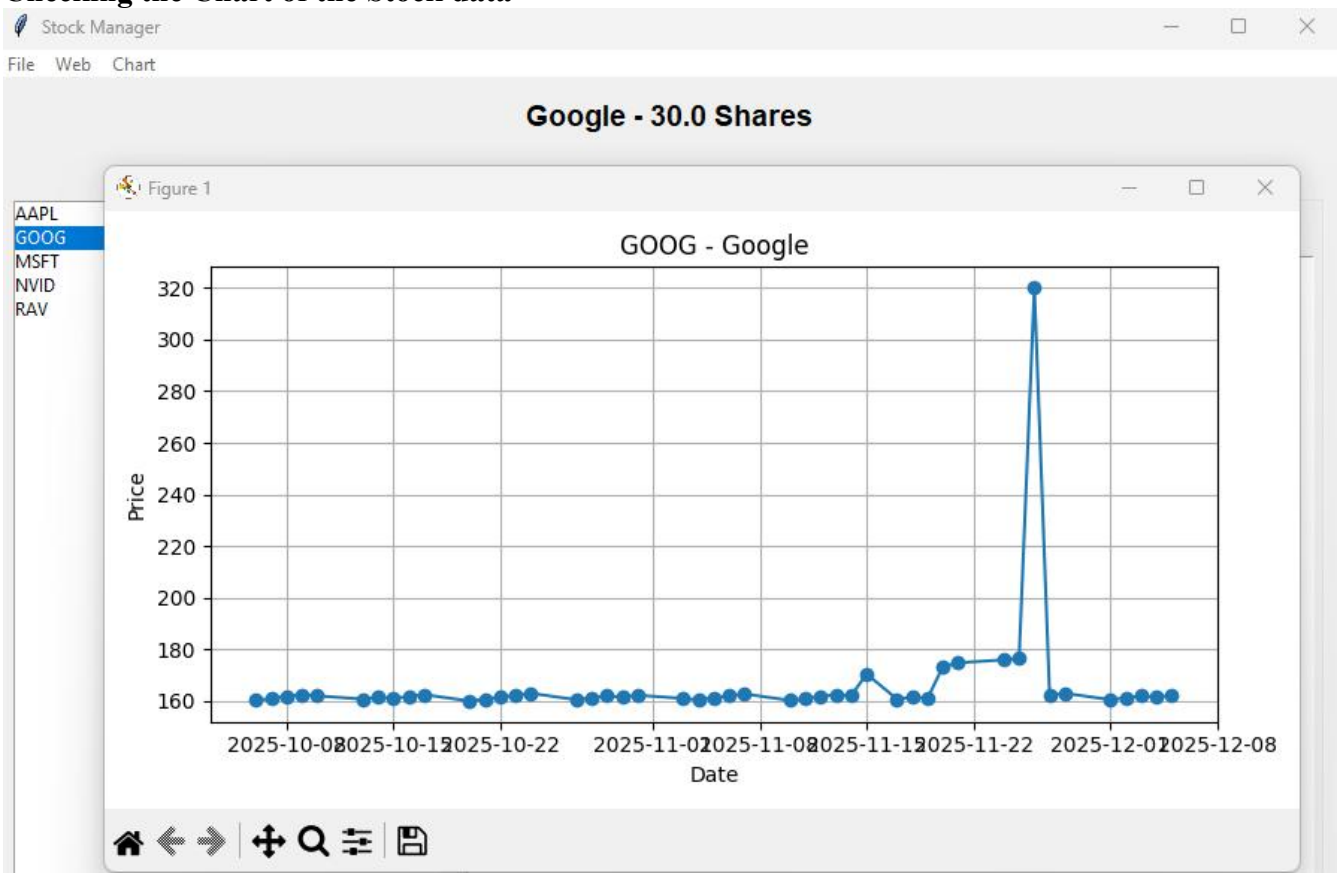


Checking whether the Daily stock data history is updated properly

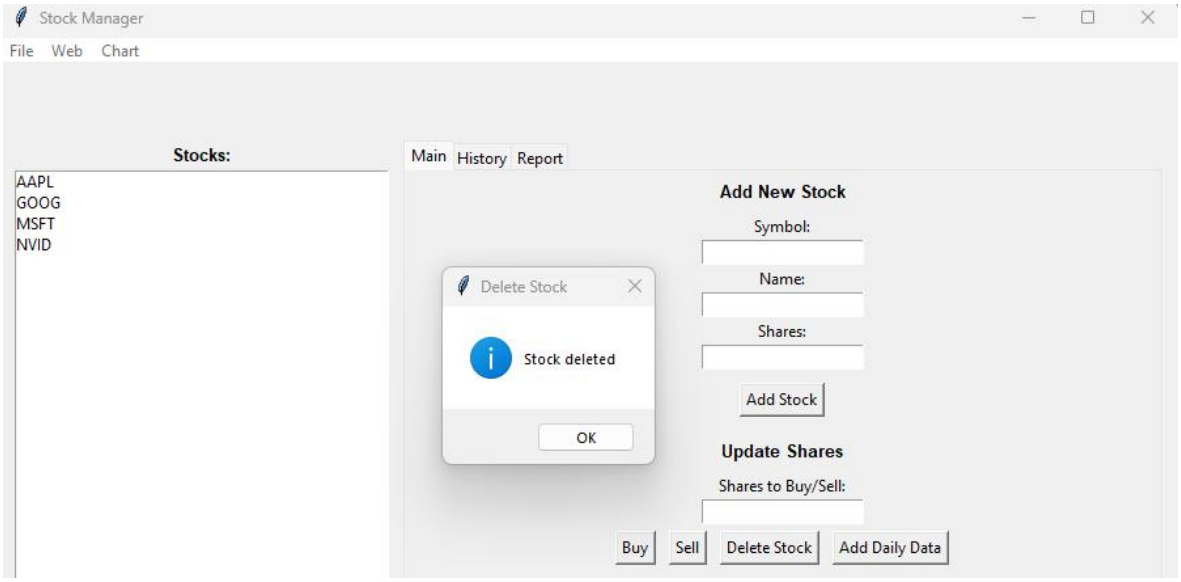
Checking whether the Share Report is updated properly



Checking the Chart of the Stock data

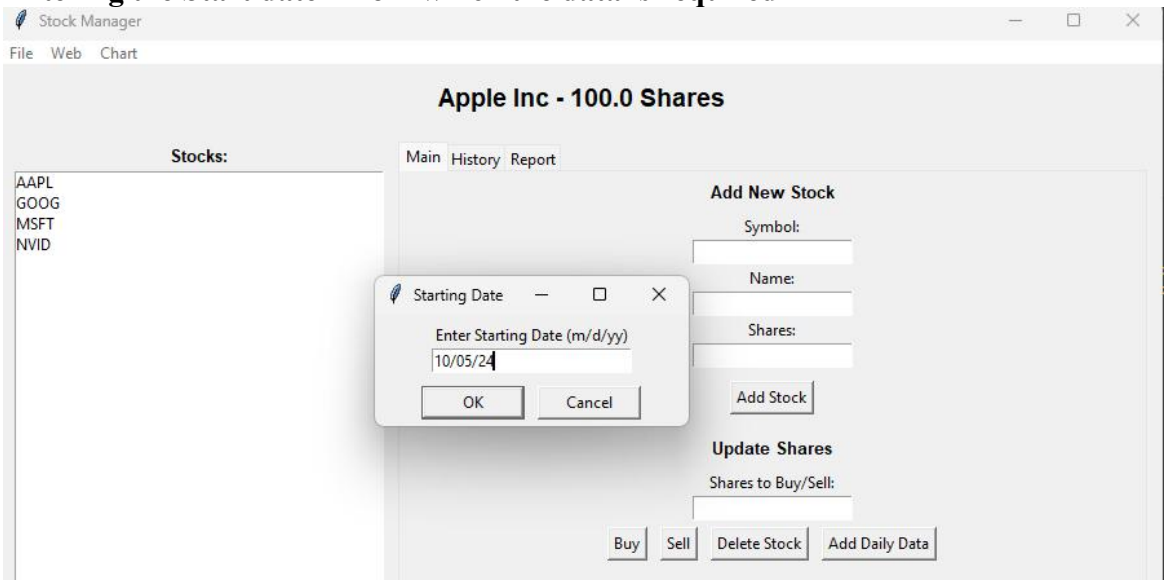


Checking the Delete Stock - by Deleting the stock RAV

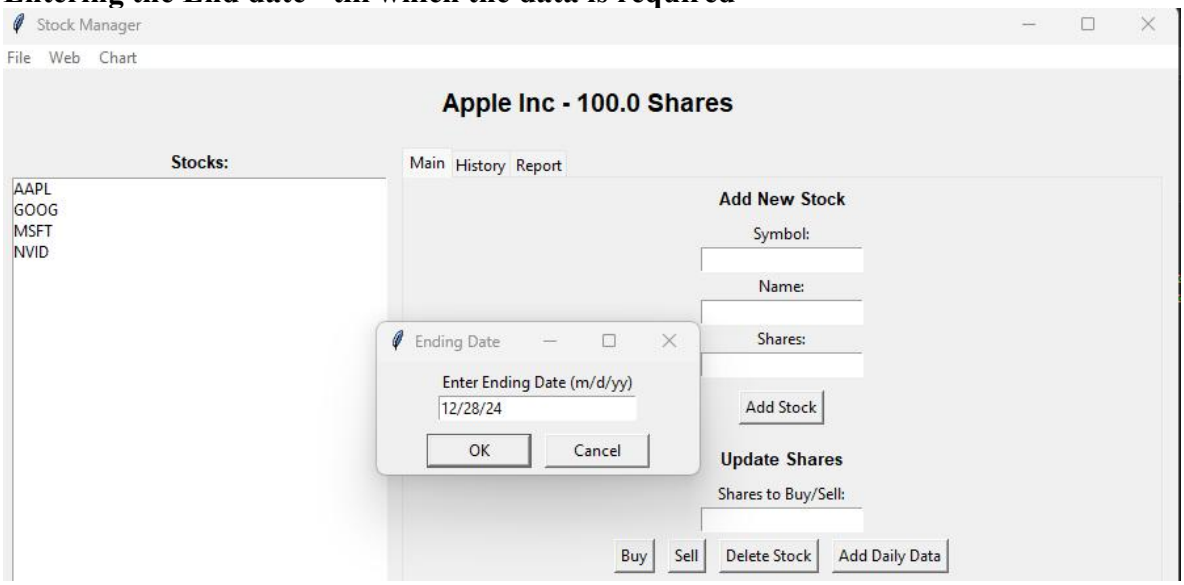


Retriving Stock from web
Select Web from the left top and select Retrive from web option

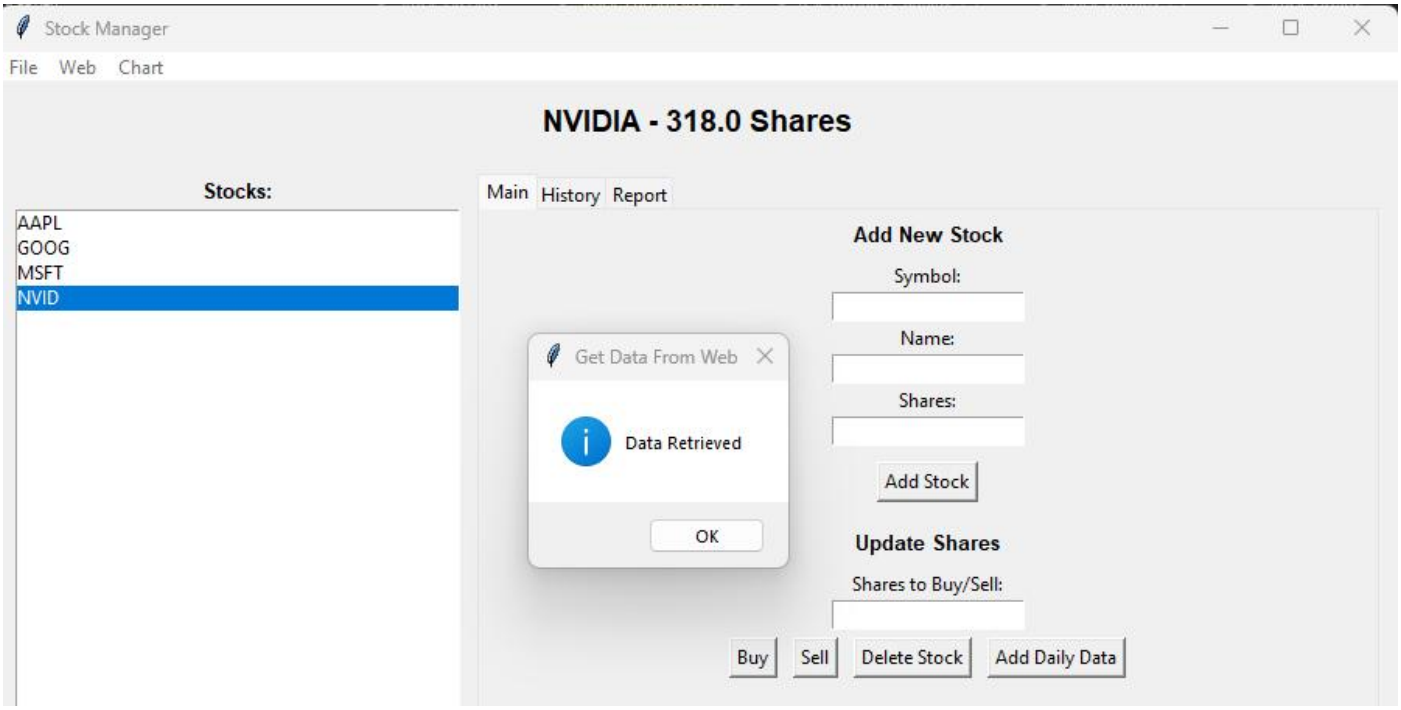
Entering the Start date - from which the data is required



Entering the End date - till which the data is required

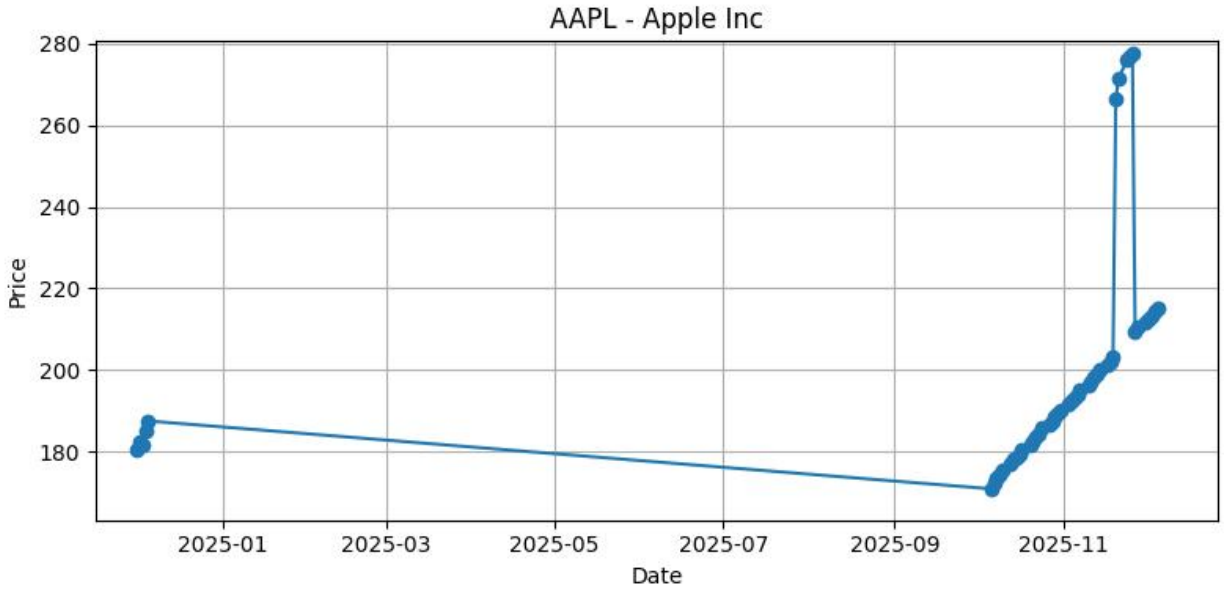


Now thw Data is retrived from the Web

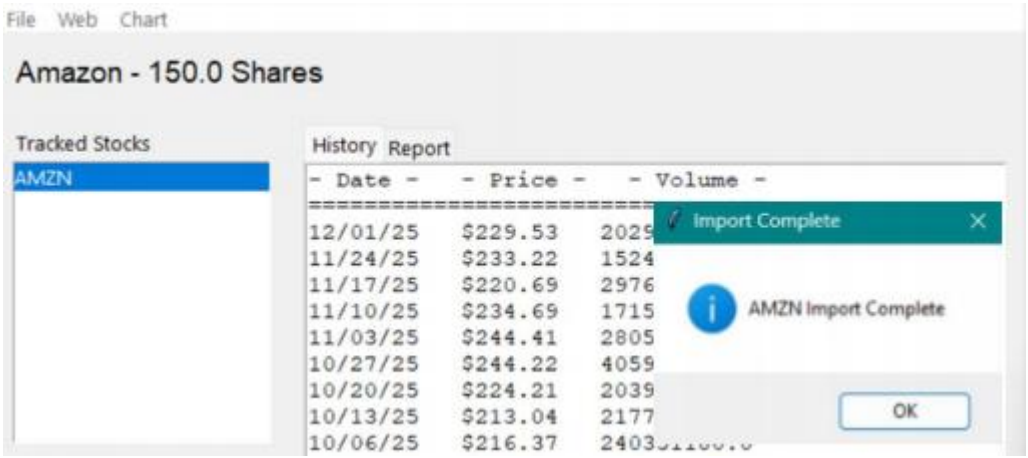


Checking whether the History shows the updated data between the date's entered

Checking whether the Chart is also updated



Importing data of a stock using csv file



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