

Name: Denisha Ketan Tank.

Lab 2

Code::

```
# Denisha TANK

from datetime import datetime

class Stock:
    def __init__(self, symbol, name, shares):
        self._symbol = symbol
        self._name = name
        self._shares = shares
        self.DataList = []

    @property
    def symbol(self):
        return self._symbol
    @symbol.setter
    def symbol(self, symbol):
        raise RuntimeWarning("Cannot Change Stock Symbol")

    @property
    def name(self):
        return self._name
    @name.setter
    def name(self, name):
        self._name = name

    @property
    def shares_(self):
        return self._shares
    @shares_.setter
    def shares_(self, shares):
        raise RuntimeWarning("Use buy() or sell() to change shares.")
```

```
denishatank / Downloads / stock_class.py / ...
def main():
    error_count = error_count + 1
    error_list.append("Buy Shares Failure!")
    print("Test Sell shares...",end="")
    try:
        testStock.sell(25)
        if testStock.shares_ == 125:
            print("Successful!")
        else:
            print("ERROR! Sell shares unsuccessful.")
            error_count = error_count+1
            error_list.append("Sell Shares Failure!")
    except:
        print("ERROR! Sell shares failed.")
        error_count = error_count + 1
        error_list.append("Sell Shares Failure!")

    # Test add daily data
    print("Creating daily stock data...",end="")
    daily_data_error = False
    try:
        dayData = DailyData(datetime.strptime("1/1/20","%m/%d/%y"),float(14.50),float(14.50))
        testStock.add_data(dayData)
        if testStock.DataList[0].date != datetime.strptime("1/1/20","%m/%d/%y"):
            error_count = error_count + 1
            daily_data_error = True
            error_list.append("Add Daily Data - Problem with Date")
        if testStock.DataList[0].close != 14.50:
            error_count = error_count + 1
            daily_data_error = True
            error_list.append("Add Daily Data - Problem with Closing Price")
        if testStock.DataList[0].volume != 100000:
            error_count = error_count + 1
            daily_data_error = True
    
```

```
#Denisha Tank

import stock_console
import stock_GUI

def main():

    stock_console.main()

    return

if __name__ == "__main__":

```

```

# Denisha Tank

from datetime import datetime
from stock_class import Stock, DailyData
from utilities import clear_screen, display_stock_chart
from os import path
import stock_data

# Main Menu
def main(stock_list):
    option = ""
    while option != "0":
        clear_screen()
        print("Stock Analyzer ---")
        print("1 - Manage Stocks (Add, Update, Delete, List)")
        print("2 - Add Daily Stock Data (Date, Price, Volume)")
        print("3 - Show Report")
        print("4 - Show Chart")
        print("5 - Manage Data (Save, Load, Retrieve)")
        print("0 - Exit Program")
        option = input("Enter Menu Option: ")
        while option not in ["1","2","3","4","5","0"]:
            clear_screen()
            print("*** Invalid Option - Try again ***")
            print("Stock Analyzer")
            print("1 - Manage Stocks (Add, Update, Delete, List)")
            print("2 - Add Daily Stock Data (Date, Price, Volume)")
            print("3 - Show Report")
            print("4 - Show Chart")
            print("5 - Manage Data (Save, Load, Retrieve)")
            print("0 - Exit Program")
            option = input("Enter Menu Option: ")
        if option == "1":

```

```

def buying_stock(stock_list):
    symbol = input("Enter Stock Symbol to Buy: ").upper()
    stocks = next((s for s in stock_list if s.symbol.upper() == symbol), None)
    if stocks is None:
        print("Symbol not found.")
        input("Press Enter to continue...")
        return
    try:
        shares = float(input("Enter Number of Shares to Buy: "))
    except ValueError:
        print("Invalid number of shares.")
        input("Press Enter to continue...")
        return
    stocks.buy(shares)
    print("Shares updated.")
    input("Press Enter to continue...")

def selling_stock(stock_list):
    clear_screen()
    print("Sell Shares ---")
    if not stock_list:
        print("No stocks in portfolio.")
        input("Press Enter to continue...")
        return
    print("Stock List: [", end="")
    print(", ".join([stock.symbol for stock in stock_list]), end="")
    print("]")
    symbol = input("Enter Stock Symbol to Sell: ").upper()
    stock = next((s for s in stock_list if s.symbol.upper() == symbol), None)
    if stock is None:
        print("Symbol not found.")
        input("Press Enter to continue...")
        return

```

```

# Denisha Tank

import sqlite3
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
import re
import pandas as pd
import os
import csv
import time
from datetime import datetime
from utilities import clear_screen
from utilities import sortDailyData
from stock_class import Stock, DailyData

# Creating the SQLite database
def create_database():
    stockDB = "stocks.db"
    conn = sqlite3.connect(stockDB)
    cur = conn.cursor()
    createStockTableCmd = """CREATE TABLE IF NOT EXISTS stocks (
        symbol TEXT NOT NULL PRIMARY KEY,
        name TEXT,
        shares REAL
    );"""
    createDailyDataTableCmd = """CREATE TABLE IF NOT EXISTS dailyData (
        symbol TEXT NOT NULL,
        date TEXT NOT NULL,
        price REAL NOT NULL,
        volume REAL NOT NULL
    );"""

```

```

def saving_stock_data(stock_list):
    cur.execute(insertStockCmd, insertValues)
    cur.execute("COMMIT;")
except:
    pass
for daily_data in stock.DataList:
    insertValues = (stock.symbol,daily_data.date.strftime("%m/%d/%y"),daily_data.close,daily_data.volume)
    try:
        cur.execute(insertDailyDataCmd, insertValues)
        cur.execute("COMMIT;")
    except:
        pass

def loading_stock_data(stock_list):
    stock_list.clear()
    stockDB = "stocks.db"
    conn = sqlite3.connect(stockDB)
    stockCur = conn.cursor()
    stockSelectCmd = """SELECT symbol, name, shares
    FROM stocks; """
    stockCur.execute(stockSelectCmd)
    stockRows = stockCur.fetchall()
    for row in stockRows:
        new_stock = Stock(row[0],row[1],row[2])
        dailyDataCur = conn.cursor()
        dailyDataCmd = """SELECT date, price, volume
        FROM dailyData
        WHERE symbol=?; """
        selectValue = (new_stock.symbol)
        dailyDataCur.execute(dailyDataCmd,(selectValue,))
        dailyDataRows = dailyDataCur.fetchall()

```

```

# Denisha Tank

import datetime
import path
import *
import ttk
import messagebox, simpledialog, filedialog
csv
stock_data
stock_class import Stock, DailyData
utilities import clear_screen, display_stock_chart, sortStocks, sortDailyData

stockApp:
__init__(self):
    self.stock_list = []

    if path.exists("stocks.db") == False:
        stock_data.create_database()

    self.root = Tk()
    self.root.title("Stock Manager") # Replace with a suitable name for your program

    self.menuBar = Menu(self.root)

    # Adding File Menu
    self.fileMenu = Menu(self.menuBar, tearoff=0)
    self.fileMenu.add_command(label="Load Data", command=self.load)

```

```

# Denisha Tank
import matplotlib.pyplot as plt

from os import system, name

def clearing_screen():
    if name == "nt":
        _ = system('cls')
    else:
        _ = system('clear')

def sortingStocks(stock_list):
    stock_list.sort(key=lambda stock: stock.symbol.upper())

def sortingofDailyData(stock_list):
    for stock in stock_list:
        stock.DataList.sort(key=lambda daily: daily.date)

def displaying_stock_chart(stock_list, symbol):
    target = next((s for s in stock_list if s.symbol.upper() == symbol), None)
    if target is None or not target.DataList:
        print("No data available to chart for symbol:", symbol)
        return

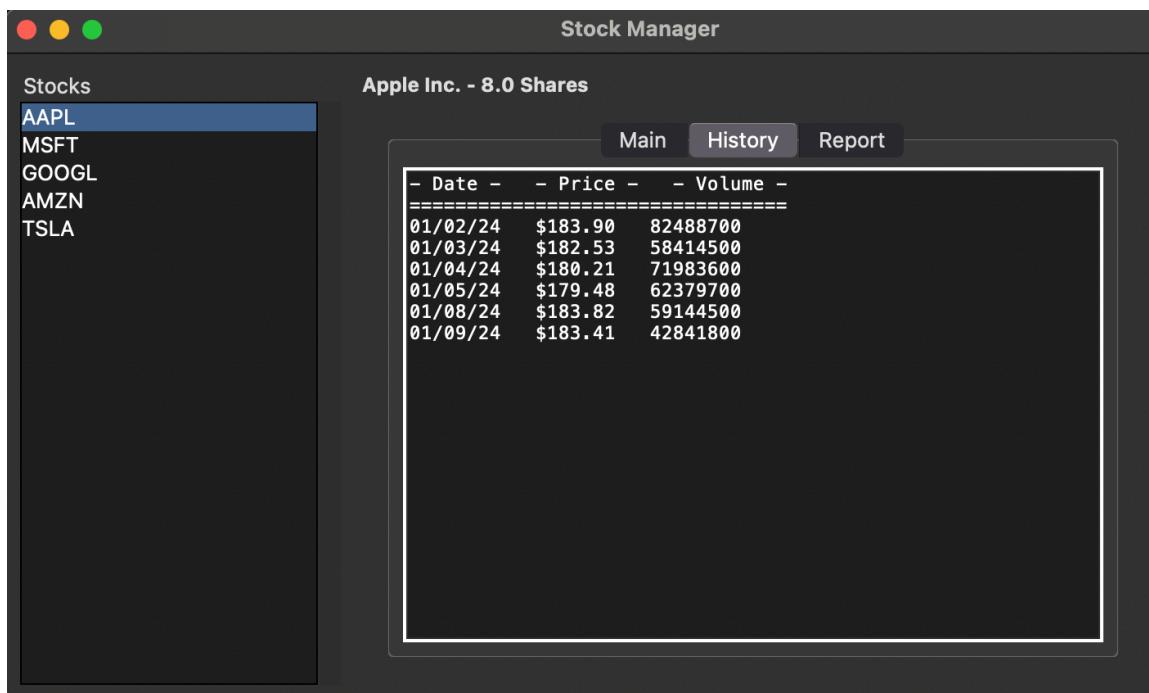
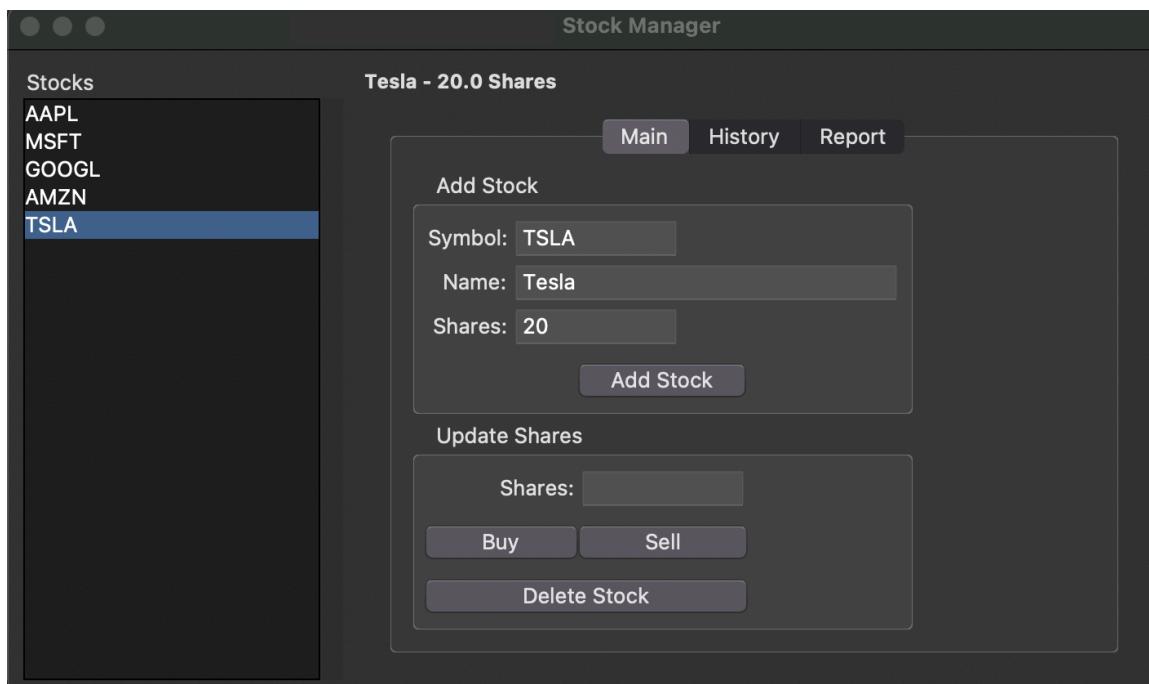
    target.DataList.sort(key=lambda d: d.date)

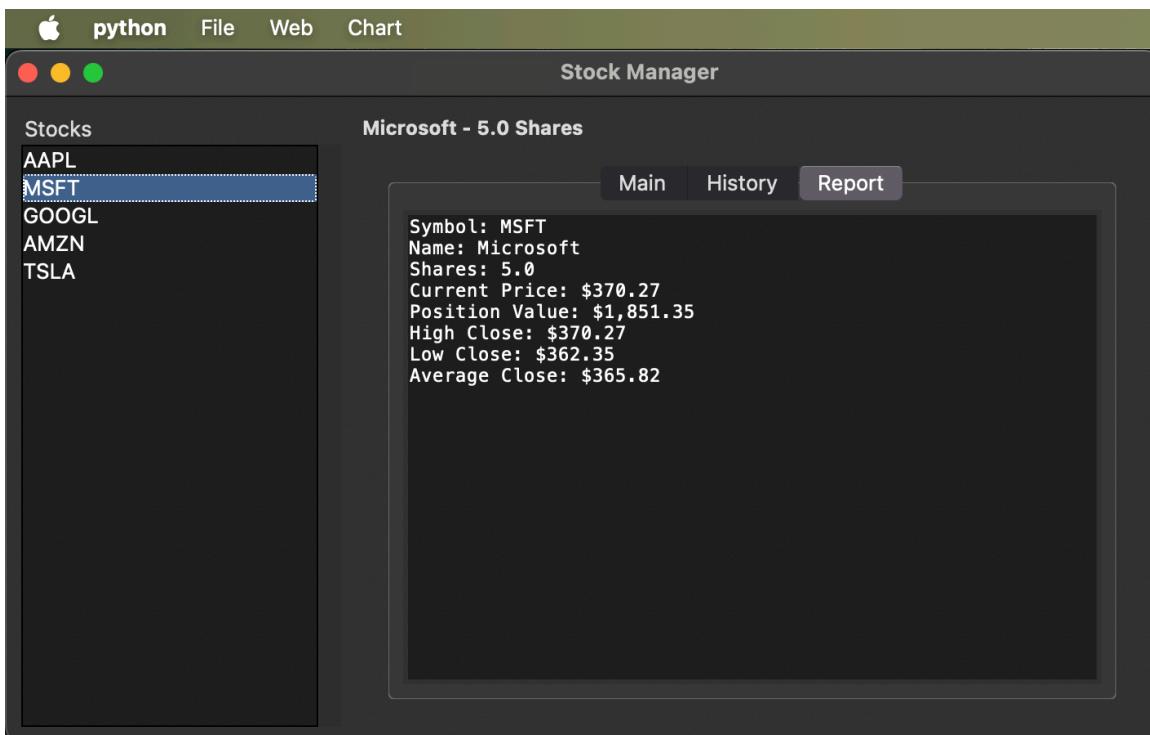
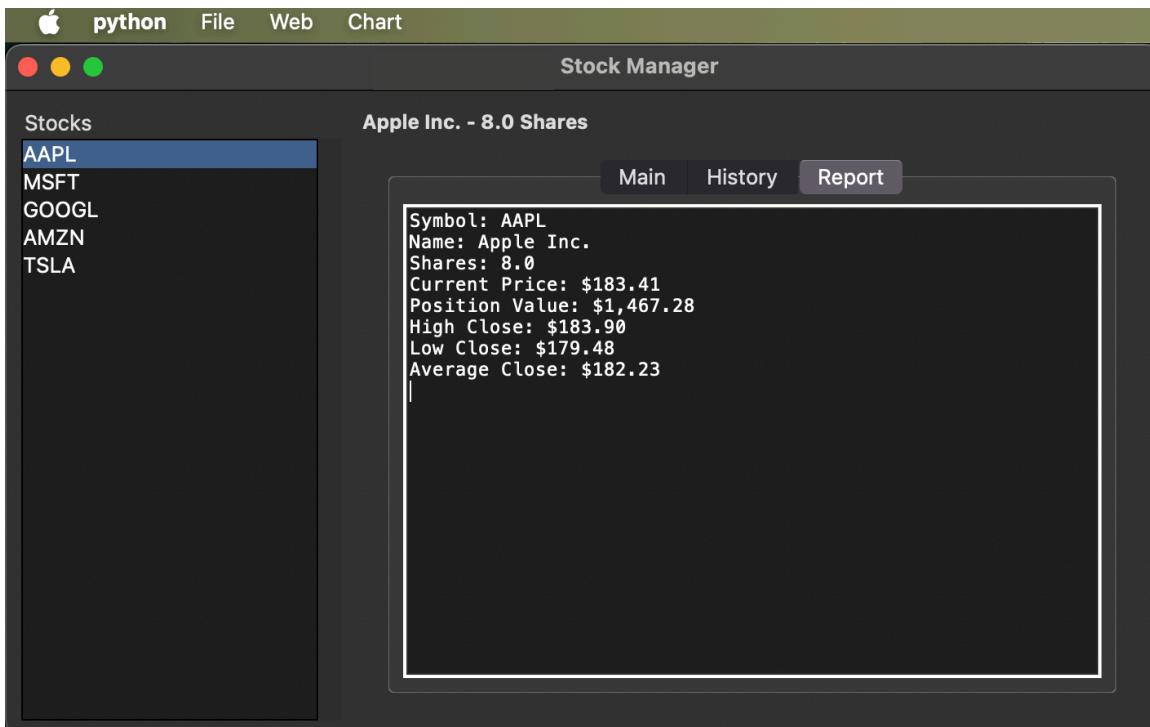
    dates = [d.date for d in target.DataList]
    closes = [d.close for d in target.DataList]

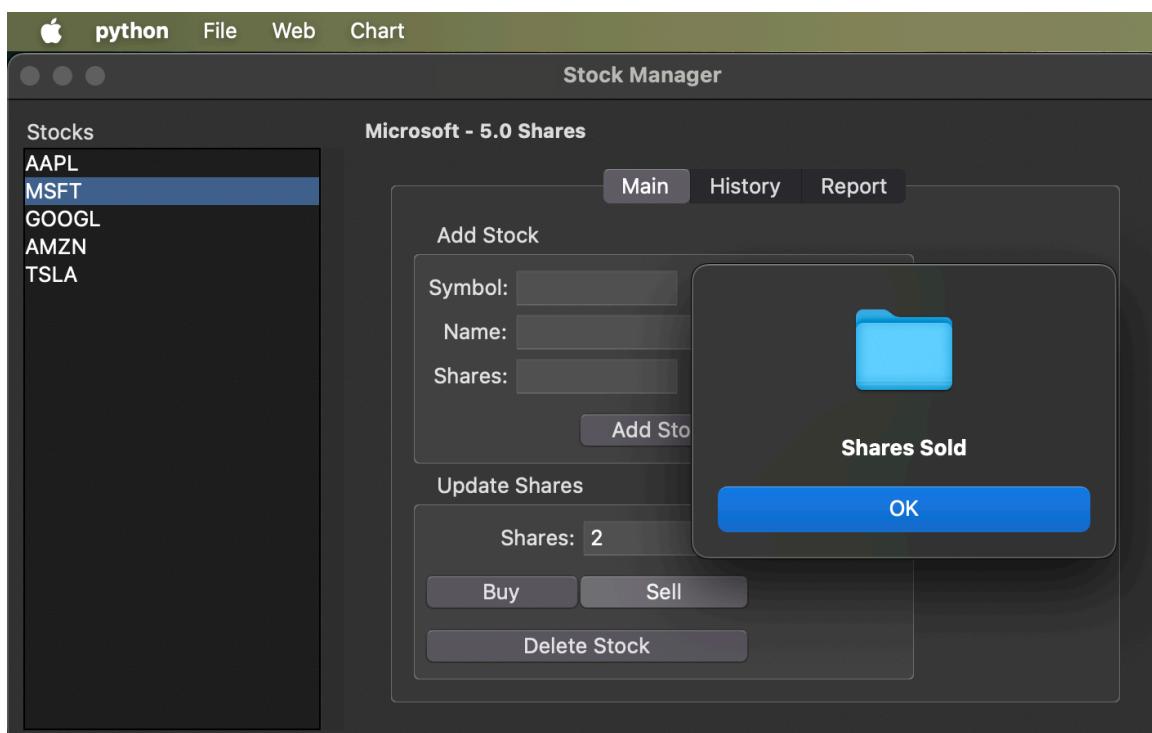
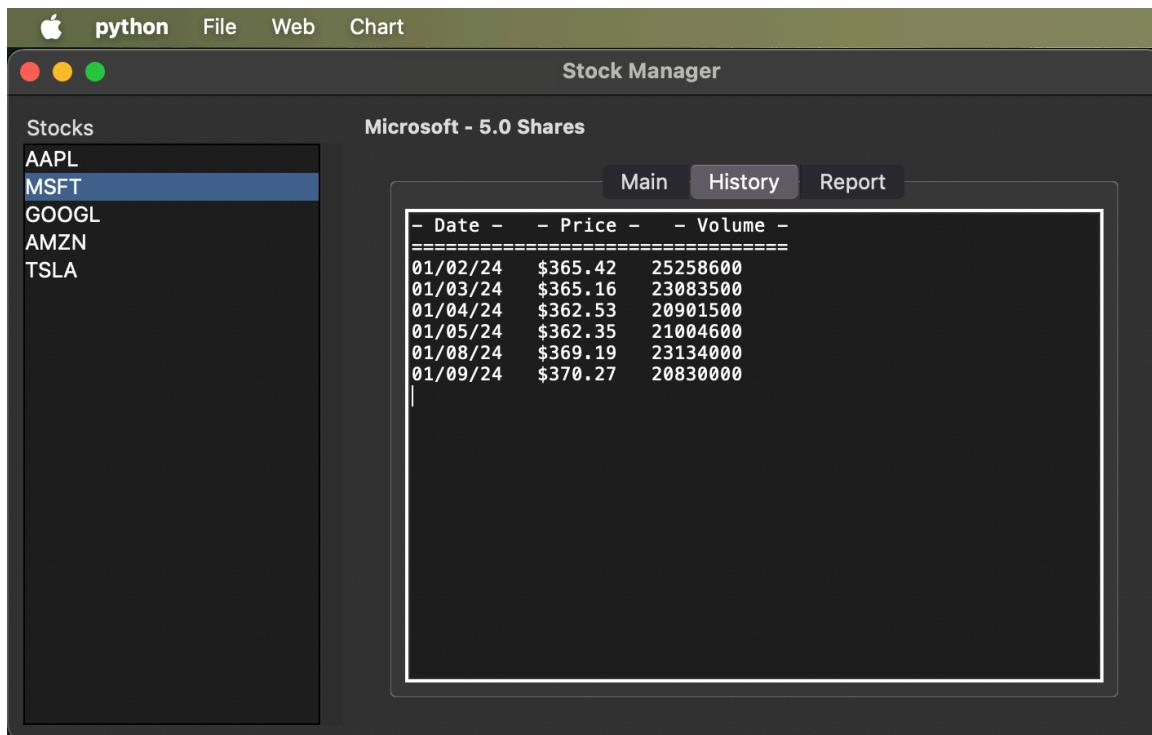
    plt.figure()
    plt.plot(dates, closes)
    plt.title(f"{target.symbol} Closing Price History")

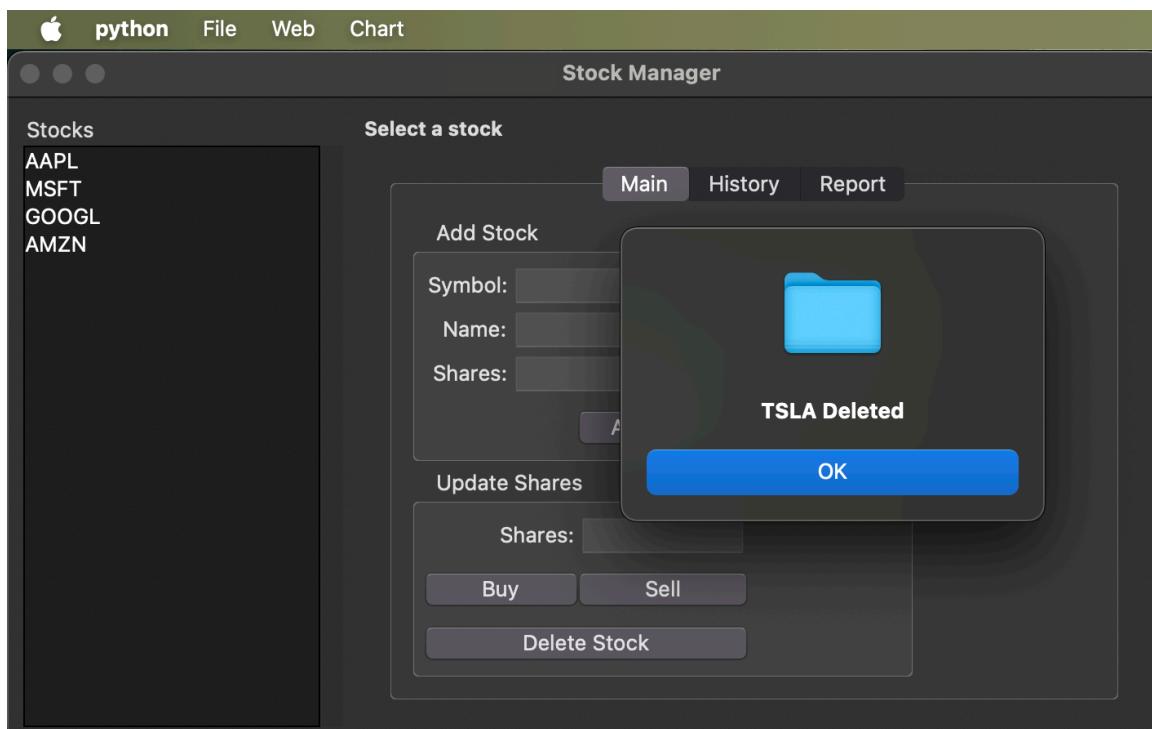
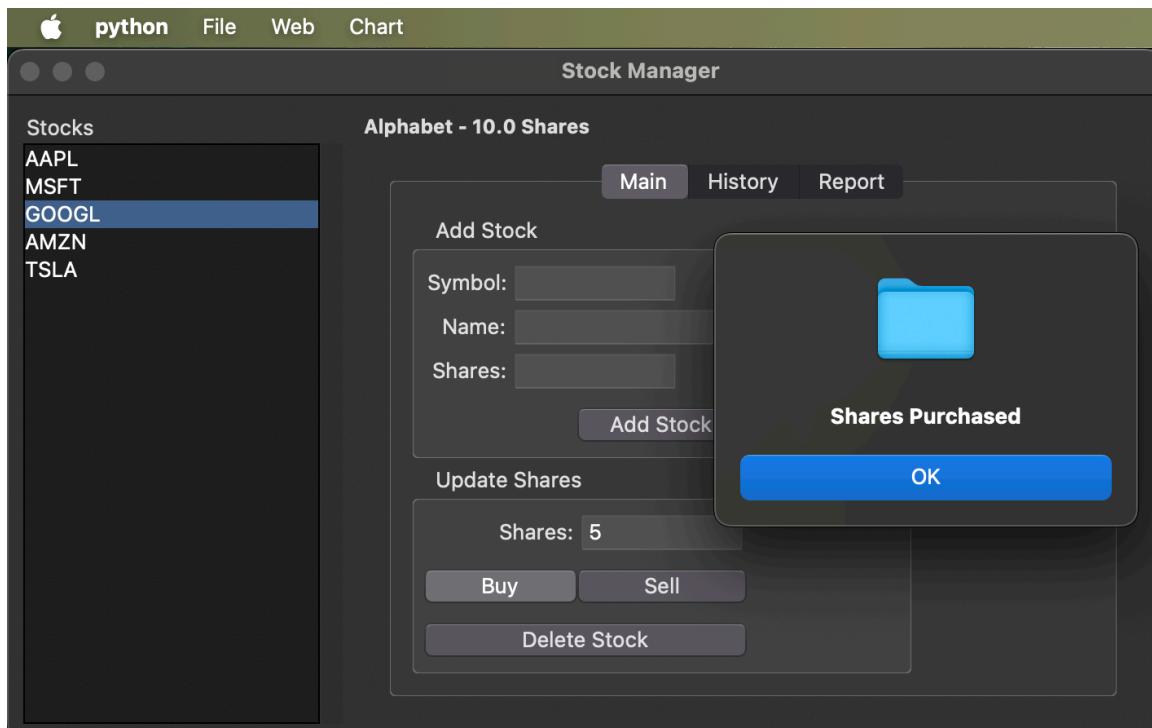
```

Output:









Stock Manager

Stocks

AAPL

MSFT

GOOGL

AMZN

Apple Inc. - 8.0 Shares

Main History Report

- Date -	- Price -	- Volume -
01/02/24	\$183.90	82488700
01/03/24	\$182.53	58414500
01/04/24	\$180.21	71983600
01/05/24	\$179.48	62379700
01/08/24	\$183.82	59144500
01/09/24	\$183.41	42841800
12/08/25	\$277.89	38211830
12/05/25	\$278.78	47265850
12/04/25	\$280.70	43989060
12/03/25	\$284.15	43538690
12/02/25	\$286.19	53669530
12/01/25	\$283.10	46587720
11/28/25	\$278.85	20135620
11/26/25	\$277.55	33431420
11/25/25	\$276.97	46914220
11/24/25	\$275.92	65585800
11/21/25	\$271.49	59030830
11/20/25	\$266.25	45823570
11/19/25	\$268.56	40424490
11/18/25	\$267.44	45677280

python File Web Chart

Stock Manager

Stocks

AAPL

MSFT

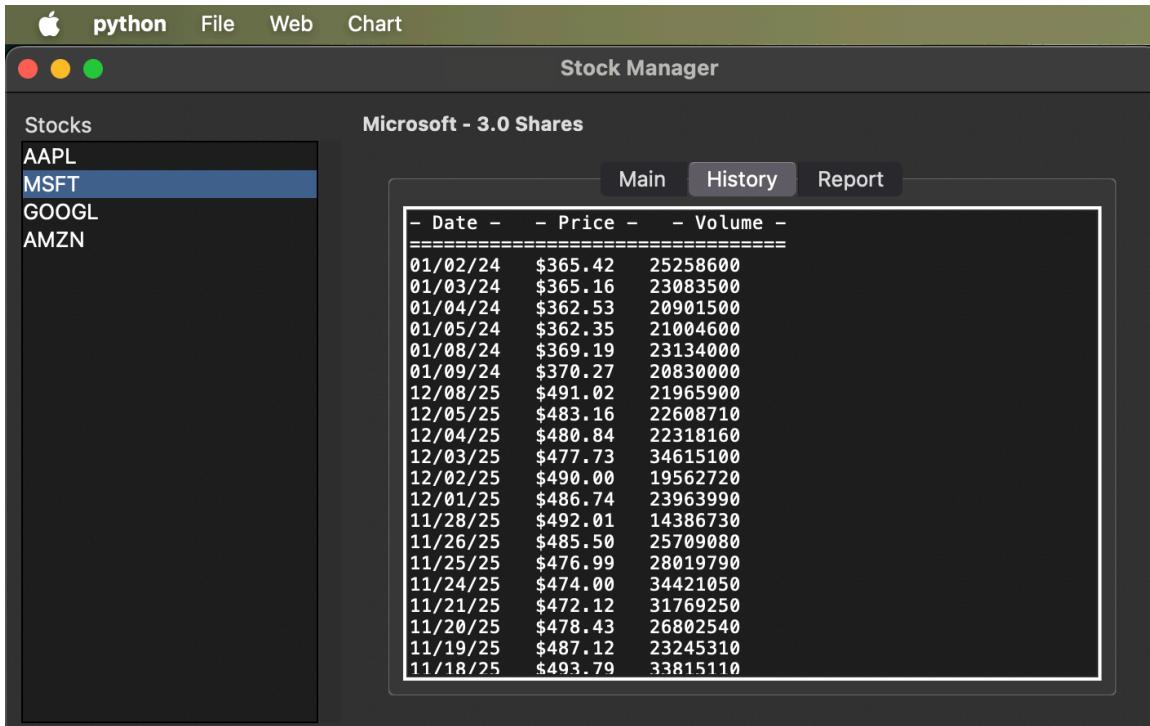
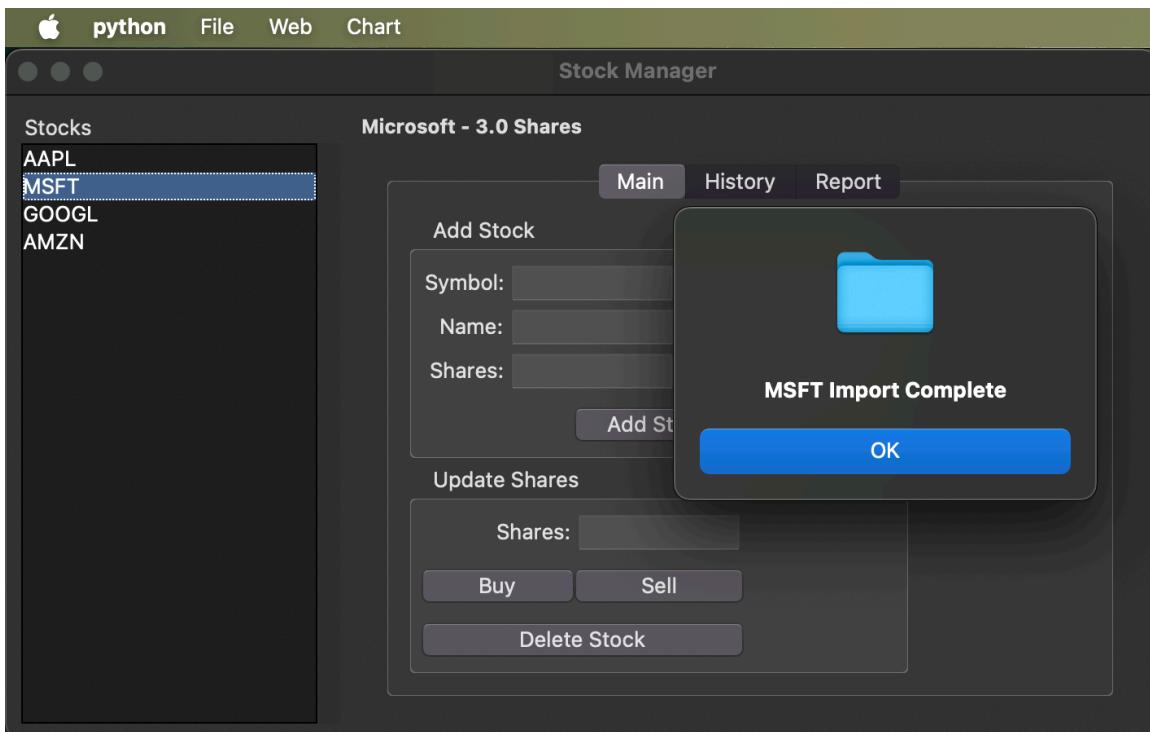
GOOGL

AMZN

Apple Inc. - 8.0 Shares

Main History Report

```
Symbol: AAPL
Name: Apple Inc.
Shares: 8.0
Current Price: $269.43
Position Value: $2,155.44
High Close: $286.19
Low Close: $179.48
Average Close: $253.78
```



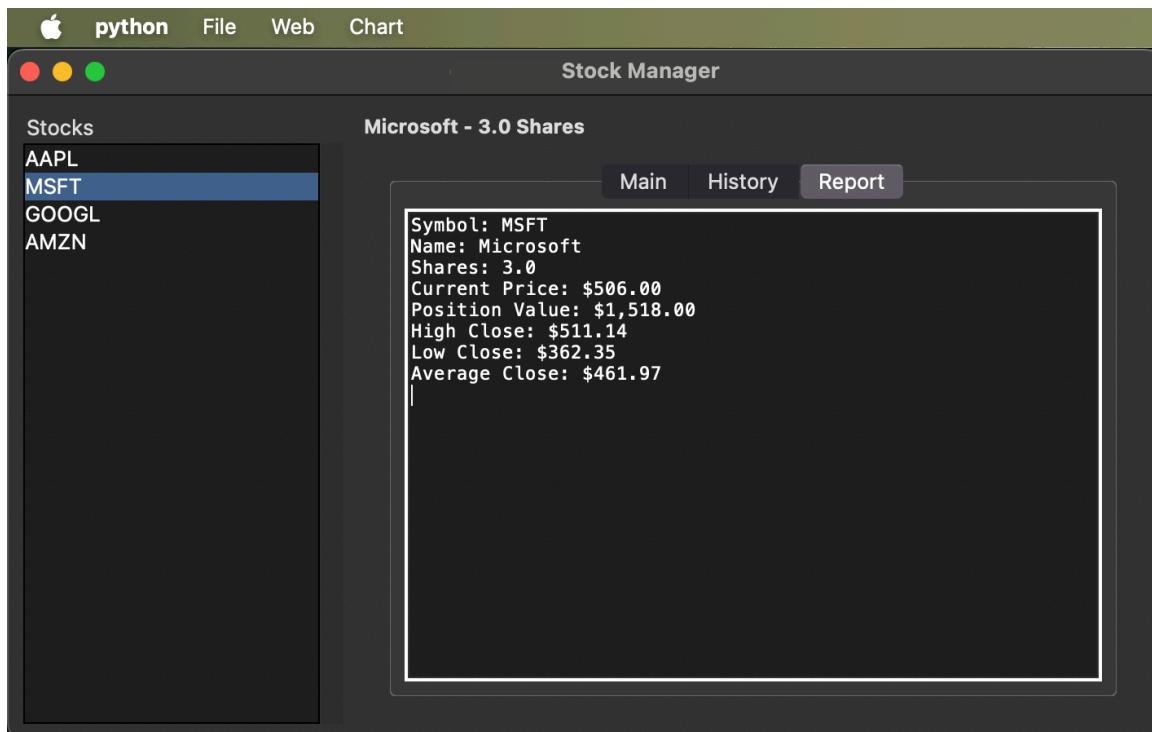


Figure 1

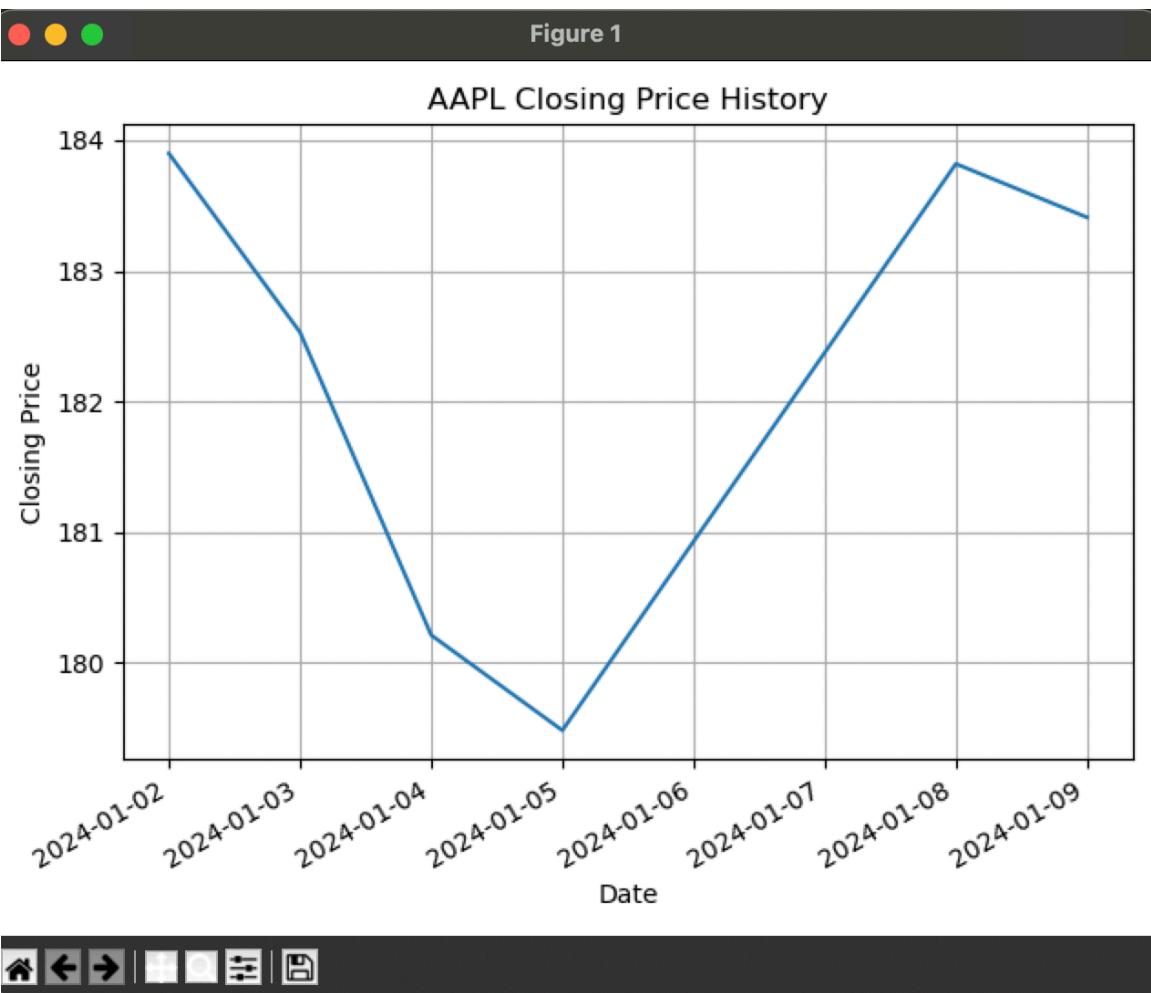
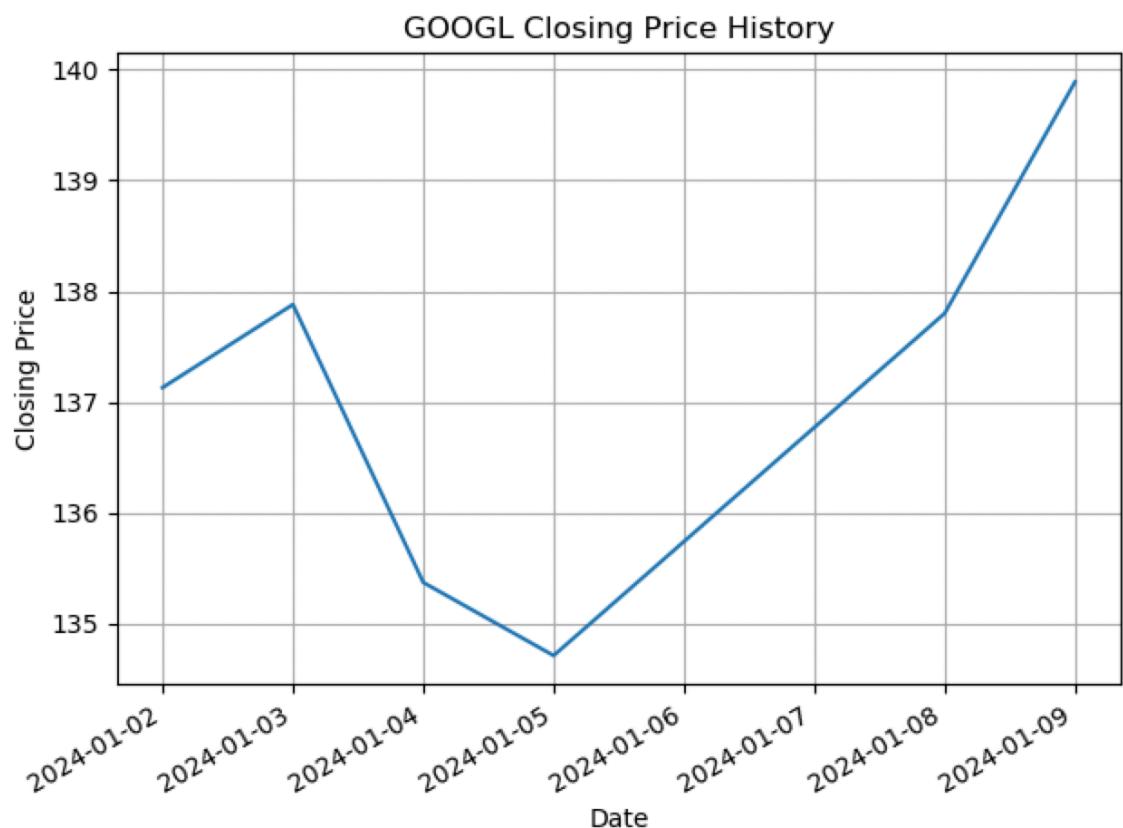
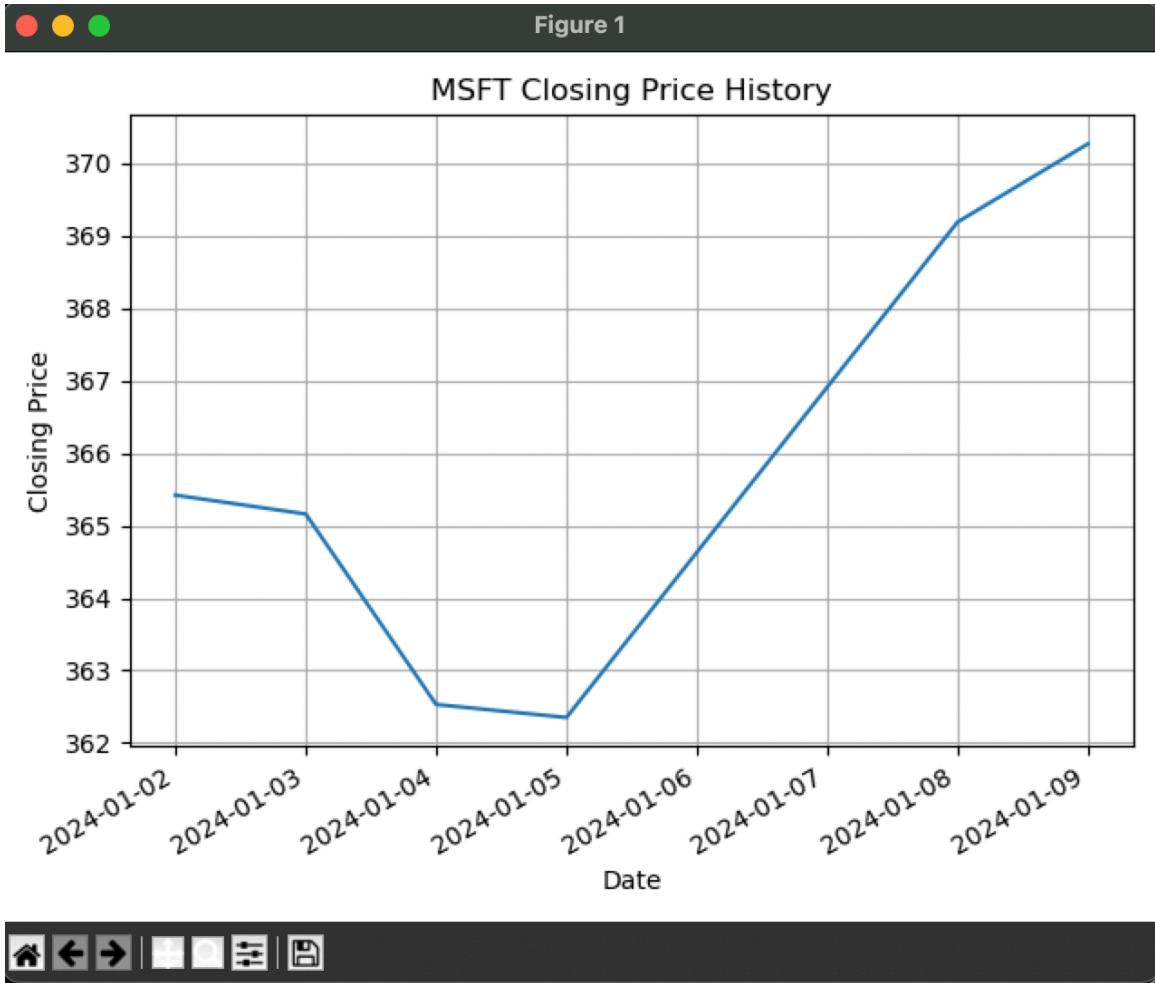




Figure 1





```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

Manage Stocks ---
1 - Add Stock
2 - Update Shares
3 - Delete Stock
4 - List Stocks
0 - Exit Manage Stocks
Enter Menu Option: 1
Add Stock ---
0 - Return to Manage Stocks
Enter Stock Symbol (or 0 to cancel): NVDA
Enter Company Name: Nvidia
Enter Number of Shares: 20
Stock Added!
Press Enter to add another stock or 0 to return: █
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

Manage Stocks ---
1 - Add Stock
2 - Update Shares
3 - Delete Stock
4 - List Stocks
0 - Exit Manage Stocks
Enter Menu Option: 4
Stocks Being Tracked ---
AAPL - Apple Inc. (8.0 shares)
MSFT - Microsoft (5.0 shares)
GOOGL - Alphabet (10.0 shares)
AMZN - Amazon (7.0 shares)
TSLA - Tesla (20.0 shares)
NVDA - Nvidia (20.0 shares)
Press Enter to continue...█
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
Buy Shares ---  
Stock List: [AAPL, MSFT, GOOGL, AMZN, TSLA, NVDA]  
Enter Stock Symbol to Buy: AAPL  
Enter Number of Shares to Buy: 3  
Shares updated.  
Press Enter to continue...█
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
Delete Stock ---  
Stock List: [AAPL, MSFT, GOOGL, AMZN, TSLA, NVDA]  
Enter Stock Symbol to Delete: TSLA  
Stock deleted.  
Press Enter to continue...█
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
Add Daily Stock Data ---  
Stock List: [AAPL, MSFT, GOOGL, AMZN, NVDA]  
Enter Stock Symbol: AMZN  
Enter Date (m/d/yy): 1/1/24  
Enter Closing Price: 300  
Enter Volume: 10  
Daily data added.  
Add another record? (Y/N): █
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

Stock Report ---

AAPL - Apple Inc. (11.0 shares)
Date Close Volume
01/02/24 183.90 82488700
01/03/24 182.53 58414500
01/04/24 180.21 71983600
01/05/24 179.48 62379700
01/08/24 183.82 59144500
01/09/24 183.41 42841800

MSFT - Microsoft (5.0 shares)
Date Close Volume
01/02/24 365.42 25258600
01/03/24 365.16 23083500
01/04/24 362.53 20901500
01/05/24 362.35 21004600
01/08/24 369.19 23134000
01/09/24 370.27 20830000

GOOGL - Alphabet (10.0 shares)
Date Close Volume
01/02/24 137.13 23711200
01/03/24 137.88 24212100
01/04/24 135.37 27137700
01/05/24 134.71 22513900
01/08/24 137.80 21404000
01/09/24 139.89 24759600

AMZN - Amazon (7.0 shares)
Date Close Volume
01/01/24 300.00 5
01/01/24 300.00 10

NVDA - Nvidia (20.0 shares)
No daily data.

Press Enter to continue...■
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

Manage Data ---
1 - Save to Database
2 - Load from Database
3 - Retrieve Data From Web
4 - Import From CSV File
0 - Return to Main Menu
Enter Menu Option: 1
Data saved to database.
Press Enter to continue...■
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    SERIAL MONITOR

Manage Data ---
1 - Save to Database
2 - Load from Database
3 - Retrieve Data From Web
4 - Import From CSV File
0 - Return to Main Menu
Enter Menu Option: 4
Import From CSV File ---
Enter Stock Symbol: AAPL
Enter CSV Filename (full path): /Users/denisha/Downloads/DenishaData_12255225129.csv
Data imported for AAPL.
Press Enter to continue...
```

Github Link: <https://github.com/denishatank12/Lab-2>