

STOCK RECOMMENDATION SYSTEM

NAME : **Mohammed Sufiyaan M Shaikh**

ADMISSION NO. : **----**

ROLL NUMBER : **----**

CLASS : **XII - B**

SUBJECT : **COMPUTER SCIENCE (CODE – 083)**

YEAR : **2022-23**

SCHOOL : **DELHI PUBLIC SCHOOL SURAT**

INTRODUCTION

This project focuses on the stock market as a whole. There are 2 major components of the project.

1. Graph Analysis
2. Portfolio Optimization

The central theme of the project is ‘**Modern Portfolio Theory**’ explained below.

The modern portfolio theory (MPT) is a practical method for selecting investments in order to maximize their overall returns within an acceptable level of risk. This mathematical framework is used to build a portfolio of investments that maximize the amount of expected return for the collective given level of risk.

American economist Harry Markowitz pioneered this theory in his paper "Portfolio Selection," which was published in the Journal of Finance in 1952. He was later awarded a Nobel Prize for his work on modern portfolio theory.

A key component of the MPT theory is diversification. Most investments are either high risk and high return or low risk and low return. Markowitz argued that investors could achieve their best results by choosing an optimal mix of the two based on an assessment of their individual tolerance to risk.

Understanding the Modern Portfolio Theory (MPT)

Modern portfolio theory argues that any given investment's risk and return characteristics should not be viewed alone but should be evaluated by how it affects the overall portfolio's risk and return. *That is, an investor can construct a portfolio of multiple assets that will result in greater returns without a higher level of risk.* As an alternative, starting with a desired level of expected return, the investor can construct a portfolio with the lowest possible risk that is capable of producing that return. *Based on statistical measures such as variance and correlation, a single investment's performance is less important than how it impacts the entire portfolio.*

MODULES UTILIZED

pandas - It is a python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

datetime - datetime in Python is **the combination between dates and times**.

The attributes of this class are similar to both date and separate classes. These attributes include day, month, year, minute, second, microsecond, hour.

bokeh - Bokeh is a Python library for creating interactive visualizations for modern web browsers. It helps you build beautiful graphics, ranging from simple plots to complex dashboards with streaming datasets.

math - The Python math module **provides functions** that allow you to calculate a range of important values, including the following: The factorials of a number. The greatest common divisor of two numbers. The sum of iterables, etc.

PROJECT CODE

MASTER.PY

```
import re
import csv
x = False

def write(a,b):
    f=open("user.csv", 'w', newline=' ')
    st=csv.writer(f,delimiter=',')
    st.writerow( ["Username", "Password"])
    l=[a,b]
    st.writerow(l)
    f.close()

def read(a,b):
    f=open("user.csv", 'r', newline=' ')
    st=csv.reader(f,delimiter=',')
    for i in st:
        if i[0] == a and i[1] == b:
            return True
    else:
        print("Incorrect Password")
        return False
    f.close()

ch=True

while ch==True:
    print("1. Sign up")
    print("2. Login")
```

```
choice=int(input("Enter  
choice"))

if choice==1:  
    username=input("Enter username")  
    m=True  
  
    while m==True:  
        valid=False  
        pwd=input("Enter password:")  
        if len(pwd)<6 or len(pwd)>16:  
            print("Not valid,check length of password")  
            continue  
        elif not re.search("[A-Z]",pwd):  
            print("Not valid")  
            continue  
        elif not re.search("[a-z]",pwd):  
            print("Not valid")  
            continue  
        elif not re.search("[0-9]",pwd):  
            print("Not valid")  
            continue  
        elif not re.search("[\$#@]",pwd):  
            print("Not valid")  
            continue  
        else:  
            valid=True  
    if valid:  print("Password  
is valid")  m=False
```

```
    write(username, pwd)
    k=input("Do you want to
continue")  if k=='yes':
        ch=True
    else:
        ch=False
        x=True

elif choice==2:
    username=input("Enter
username")
    password=input("Enter password")
    x = read(username,password)
    ch=False

else:   print("Wrong
Choice")

if x==True:
    import visualizer
    import optimization
    print('1.Pull the latest list of stocks from the S&P
500')  print('2.Visualize any given stock')
    print('3.Get a personalized portfolio optimization')
choice = int(input('select a choice: '))

if choice == 1:
```

```
print("S&P 500")
visualizer.stocks() elif
choice == 2:

m=True
while
m==True:
    x=input("Please enter a ticker : ")
    print('I.Daily Time Series')
    print('II.Weekly Time Series')
    print('III.Monthly Time Series')

    ch2 = int(input('Please select a choice:
')) if ch2 == 1: visualizer.daily(x)
elif ch2 == 2:
    visualizer.weekly(x)
elif ch2 == 3:
    visualizer.monthly(x)
else:
    print("Wrong Choice")

ask=input("Do you want to go back to previous
menu?")
if ask.lower()=='no':
    m=False
else:
    m=True
elif choice == 3:
    optimization.opt()
```

```
else:    print("Wrong choice was  
        entered")  
  
else:  
    print("Not signed up")
```

VISUALIZER.PY

```
#Importing Libraries
import requests
import pandas as pd
import datetime
from bokeh.plotting import figure, show,
output_file import math
def stocks():
    URL =
'https://en.wikipedia.org/wiki/List_of_S%26P_500_companies'
    d = {}
    comp = pd.read_html(URL)[0]['Security'].tolist()
    tickers = pd.read_html(URL)[0]['Symbol'].tolist()
    for i in range(len(comp)):
        d[comp[i]] = tickers[i]
    for i in d:
        print(i+
': '+d[i])
def daily(x):
    #Code to obtain trade data for
    AAPL_API_KEY = 'QQDH0IXG7JD6KKL9'
    stock_name = x
    r =
requests.get('https://www.alphavantage.co/query?function=TIME_
SERIES_DAILY_ADJUSTED&symbol=' + stock_name + '&apikey=' +
API_KEY)
    result =
r.json()
    dataForAllDays = result['Time Series (Daily)']
    #convert to dataframe
    df = pd.DataFrame.from_dict(dataForAllDays,
```

```

orient='index') df = df.reset_index() #rename columns
df = df.rename(index=str, columns={"index": "date",
"1.
open": "open", "2. high": "high", "3. low": "low", "4.
close": "close", "6. volume": "volume"}) #Changing to datetime
df['date'] = pd.to_datetime(df['date'])
#Sort according to date df =
df.sort_values(by=['date'])
#Changing the datatype df.open
= df.open.astype(float) df.close
= df.close.astype(float) df.high
= df.high.astype(float) df.low =
df.low.astype(float) df.volume =
df.volume.astype(int)
#check the data
df.head()
#Check the datatype
df.info()
inc = df.close > df.open dec =
df.open > df.close w =
12*60*60*1000 # half day in ms
TOOLS = "pan,wheel_zoom,box_zoom,reset,save"
title = stock_name + ' Chart'
p = figure(x_axis_type="datetime", tools=TOOLS,
plot_width=1000, title = title)
p.xaxis.major_label_orientation = 3.14/4
p.grid.grid_line_alpha=0.3
p.segment(df.date, df.high, df.date, df.low,
color="black")

```

```

    p.vbar(df.date[inc], w, df.open[inc], df.close[inc],
fill_color="#D5E1DD", line_color="black")
    p.vbar(df.date[dec], w, df.open[dec], df.close[dec],
fill_color="#F2583E", line_color="black")
#Store as a HTML file
output_file("stock_information.html",
title="candlestick.py"
example") # Display in
browser show(p)

def weekly(x):
    #Code to obtain trade data for
    AAPL API_KEY = 'QQDH0IXG7JD6KKL9'
    stock_name = x r =
    requests.get('https://www.alphavantage.co/query?function=TIME_
SERIES_WEEKLY&symbol=' + stock_name + '&apikey=' + API_KEY)
    result = r.json()
    dataForAllDays = result['Weekly Time Series']
    #convert to dataframe
    df = pd.DataFrame.from_dict(dataForAllDays,
orient='index') df = df.reset_index() #rename columns
    df = df.rename(index=str, columns={"index": "date",
"1.
open": "open", "2. high": "high", "3. low": "low", "4.
close": "close", "5. volume": "volume"})
    #Changing to datetime
    df['date'] = pd.to_datetime(df['date'])
    #Sort according to date df =
    df.sort_values(by=['date'])

```

```

#Changing the datatype df.open =
df.open.astype(float) df.close =
df.close.astype(float) df.high =
df.high.astype(float) df.low =
df.low.astype(float) df.volume =
df.volume.astype(int)

#check the data
df.head() #Check
the datatype
df.info()

inc = df.close > df.open dec = df.open >
df.close w = 12*60*60*1000 # half day in ms
TOOLS = "pan,wheel_zoom,box_zoom,reset,save"
title = stock_name + ' Chart'

p = figure(x_axis_type="datetime", tools=TOOLS,
plot_width=1000, title = title)
p.xaxis.major_label_orientation = 3.14/4
p.grid.grid_line_alpha=0.3
p.segment(df.date, df.high, df.date, df.low,
color="black")

p.vbar(df.date[inc], w, df.open[inc], df.close[inc],
fill_color="#D5E1DD", line_color="black")
p.vbar(df.date[dec], w, df.open[dec], df.close[dec],
fill_color="#F2583E", line_color="black")

#Store as a HTML file
output_file("stock_information.html",
title="candlestick.py

example") # Display in
browser show(p)

def monthly(x):

```

```

#Code to obtain trade data for
AAPL API_KEY = 'QQDH0IXG7JD6KKL9'
stock_name = x r =
requests.get('https://www.alphavantage.co/query?function=TIME_
SERIES_MONTHLY&symbol=' + stock_name + '&apikey=' + API_KEY)
result = r.json()
dataForAllDays = result['Monthly Time Series']
#convert to dataframe
df = pd.DataFrame.from_dict(dataForAllDays,
orient='index') df = df.reset_index() #rename columns
df = df.rename(index=str, columns={"index": "date",
"1.
open": "open", "2. high": "high", "3. low": "low", "4.
close": "close", "5. volume": "volume"}) #Changing to datetime
df['date'] = pd.to_datetime(df['date'])
#Sort according to date df =
df.sort_values(by=['date'])
#Changing the datatype df.open
= df.open.astype(float) df.close
= df.close.astype(float) df.high
= df.high.astype(float) df.low =
df.low.astype(float) df.volume =
df.volume.astype(int)

#check the data
df.head() #Check
the datatype
df.info()

```

```
inc = df.close > df.open  dec = df.open >
df.close  w = 12*60*60*1000 # half day in ms
TOOLS = "pan,wheel_zoom,box_zoom,reset,save"
title = stock_name + ' Chart'
p = figure(x_axis_type="datetime", tools=TOOLS,
plot_width=1000, title = title)
p.xaxis.major_label_orientation = 3.14/4
p.grid.grid_line_alpha=0.3
p.segment(df.date, df.high, df.date, df.low,
color="black")
p.vbar(df.date[inc], w, df.open[inc], df.close[inc],
fill_color="#D5E1DD", line_color="black")
p.vbar(df.date[dec], w, df.open[dec], df.close[dec],
fill_color="#F2583E", line_color="black")
#Store as a HTML file
output_file("stock_information.html",
title="candlestick.py
example") # Display in
browser show(p)
```

OPTIMIZATION.PY

```
def opt():

    import pandas_datareader.data as
    web    import datetime    import pandas
    as pd
    pd.set_option('display.max_columns',
    None)    pd.set_option('display.max_rows',
    None)    start =
    datetime.datetime(2022,2,25)    end =
    datetime.datetime(2022,11,1)

    def get_stock(ticker):      #This function is used to fetch
the stock data for the given time period    data =
    web.DataReader(f'{ticker}', "yahoo", start, end)
    data[f'{ticker}'] = data["Close"]    data =
    data[[f'{ticker}']]    return data

    from functools import reduce

    def combine_stocks(tickers):      #This is an extension of
the get_stock funtion, it pulls real-time data for multiple
stocks"
        data_frames = []
        for i in tickers:
            data_frames.append(get_stock(i))

        df_merged = reduce(lambda  left,right:
pd.merge(left,right,on=['Date'], how='outer'), data_frames)

        return df_merged
```

```
stocks = []
alpha = eval(input("Please enter a list of 10 stock
tickers: "))
stocks =
list(alpha)
portfolio = combine_stocks(stocks)
portfolio.to_csv("portfolio.csv", index=False)
portfolio = pd.read_csv("portfolio.csv")    from
pypfopt.expected_returns import
mean_historical_return    from pypfopt.risk_models
import CovarianceShrinkage

mu = mean_historical_return(portfolio)
S = CovarianceShrinkage(portfolio).ledoit_wolf()
print(S)
from pypfopt.efficient_frontier import EfficientFrontier

ef = EfficientFrontier(mu, S)
weights = ef.max_sharpe()

cleaned_weights = ef.clean_weights()
print(dict(cleaned_weights))
ef.portfolio_performance(verbose=True)
from pypfopt.discrete_allocation import
DiscreteAllocation, get_latest_prices

latest_prices = get_latest_prices(portfolio)
da = DiscreteAllocation(weights, latest_prices,
total_portfolio_value=100000)
```

```
allocation, leftover = da.greedy_portfolio()
print("Discrete allocation:", allocation)
print("Funds remaining: ${:.2f}".format(leftover))
```

OUTPUT

MASTER.PY

USING CHOICE 1

```
In [2]: runfile('/Users/sufiyaan/Desktop/Project class 12 CS')
1. Sign up
2. Login

Enter choice1

Enter usernameVishesh

Enter password:Vish123*$
Password is valid

Do you want to continueno
1.Pull the latest list of stocks from the S&P 500
2.Visualize any given stock
3.Get a personalized portfolio optimization

select a choice:
```

USING CHOICE 2

```
In [1]: runfile('/Users/sufiyaan/Desktop/Project class 12 CS')
1. Sign up
2. Login

Enter choice2

Enter usernameVishesh

Enter passwordVish123*$
1.Pull the latest list of stocks from the S&P 500
2.Visualize any given stock
3.Get a personalized portfolio optimization

select a choice:
```

Now on selecting Choice 1 and/or 2, the chain of progression moves to “VISUALIZER.PY”

Selecting Choice 1 gives the user the “S&P 500”

The S&P 500 is an exhaustive list of 500 of America's best performing stocks.

```
1.Pull the latest list of stocks from the S&P 500
2.Visualize any given stock
3.Get a personalized portfolio optimization

select a choice: 1
S&P 500
```

Accenture: ACN
 Activision Blizzard: ATVI
 ADM: ADM
 Adobe Inc.: ADBE
 ADP: ADP
 Advance Auto Parts: AAP
 AES Corporation: AES
 Aflac: AFL
 Agilent Technologies: A
 Air Products and Chemicals: APD
 Akamai: AKAM
 Alaska Air Group: ALK
 Albermarle Corporation: ALB
 Alexandria Real Estate Equities: ARE
 Align Technology: ALGN
 Allegion: ALLE
 Alliant Energy: LNT
 Allstate: ALL
 Alphabet Inc. (Class A): GOOGL
 Alphabet Inc. (Class C): GOOG
 Altria: MO
 Amazon: AMZN
 Amcor: AMCR
 AMD: AMD
 Ameren: AEE
 American Airlines Group: AAL
 American Electric Power: AEP
 American Express: AXP
 American International Group: AIG
 American Tower: AMT
 American Water Works: AWK
 Ameriprise Financial: AMP
 AmerisourceBergen: ABC
 Ametek: AME
 Amgen: AMGN
 Amphenol: APH
 Analog Devices:ADI
 Ansys: ANSS
 Aon: AON
 APA Corporation: APA
 Apple Inc.: AAPL
 Applied Materials: AMAT
 Aptiv: APTV
 Arch Capital Group: ACGL
 Arista Networks: ANET
 Arthur J. Gallagher & Co.: AJG
 Assurant: AIZ
 AT&T: T
 Atmos Energy: ATO
 Autodesk: ADSK

AutoZone: AZO
 AvalonBay Communities: AVB
 Avery Dennison: AVY
 Baker Hughes: BKR
 Ball Corporation: BALL
 Bank of America: BAC
 Bath & Body Works, Inc.: BBWI
 Baxter International: BAX
 Bechtel Dickinson: BDX
 Berkley: WRB
 Berkshire Hathaway: BRK.B
 Best Buy: BBY
 Bio-Rad: BIO
 Bio-Techne: TECH
 Biogen: BIIB
 BlackRock: BLK
 BNY Mellon: BK
 Boeing: BA
 Booking Holdings: BKNG
 BorgWarner: BWA
 Boston Properties: BXP
 Boston Scientific: BSX
 Bristol Myers Squibb: BMY
 Broadcom Inc.: AVGO
 Broadridge Financial Solutions: BR
 Brown & Brown: BRO
 Brown-Forman: BF.B
 C.H. Robinson: CHRW
 Cadence Design Systems: CDNS
 Caesars Entertainment: CZR
 Camden Property Trust: CPT
 Campbell Soup Company: CPB
 Capital One: COF
 Cardinal Health: CAH
 CarMax: KMX
 Carnival: CCL
 Carrier Global: CARR
 Catalent: CTLT
 Caterpillar Inc.: CAT
 Cboe Global Markets: CBOE
 CBRE Group: CBRE
 CDW: CDW
 Celanese: CE
 Centene Corporation: CNC
 CenterPoint Energy: CNP
 Ceridian: CDAY
 CF Industries: CF
 Charles River Laboratories: CRL
 Charles Schwab Corporation: SCHW
 Charter Communications: CHTR

Chevron Corporation: CVX
 Chipotle Mexican Grill: CMG
 Chubb Limited: CB
 Church & Dwight: CHD
 Cigna: CI
 Cincinnati Financial: CINF
 Cintas: CTAS
 Cisco: CSC
 Citigroup: C
 Citizens Financial Group: CFG
 Clorox: CLX
 CME Group: CME
 CMS Energy: CMS
 The Coca-Cola Company: KO
 Cognizant: CTSH
 Colgate-Palmolive: CL
 Comcast: CMCSA
 Comerica: CMA
 Conagra Brands: CAG
 ConocoPhillips: COP
 Consolidated Edison: ED
 Constellation Brands: STZ
 Constellation Energy: CEG
 CooperCompanies: COO
 Copart: CPRT
 Corning Inc.: GLW
 Corteva: CTVA
 CoStar Group: CSGP
 Costco: COST
 Coterra: CTRA
 Crown Castle: CCI
 CSX: CSX
 Cummins: CMI
 CVS Health: CVS
 D.R. Horton: DHI
 Danaher Corporation: DHR
 Darden Restaurants: DRI
 DaVita Inc.: DVA
 John Deere: DE
 Delta Air Lines: DAL
 Dentsply Sirona: XRAY
 Devon Energy: DVN
 Dexcom: DXCM
 Diamondback Energy: FANG
 Digital Realty: DLR
 Discover Financial: DFS
 Dish Network: DISH
 Disney: DIS
 Dollar General: DG
 Dollar Tree: DLTR

Ford Motor Company: F
 Fortinet: FTNT
 Fortive: FTV
 Fox Corporation (Class A): FOXA
 Fox Corporation (Class B): FOX
 Franklin Templeton: BEN
 Freeport-McMoRan: FCX
 Garmin: GRMN
 Gartner: IT
 Gen Digital Inc.: GEN
 Generac: GNRC
 General Dynamics: GD
 General Electric: GE
 General Mills: GIS
 General Motors: GM
 Genuine Parts Company: GPC
 Gilead Sciences: GILD
 Globe Life: GL
 Global Payments: GPN
 Goldman Sachs: GS
 Halliburton: HAL
 Hartford (The): HIG
 Hasbro: HAS
 HCA Healthcare: HCA
 Healthpeak: PEAK
 Henry Schein: HSIC
 Hershey's: HSY
 Hess Corporation: HES
 Hewlett Packard Enterprise: HPE
 Hilton Worldwide: HLT
 Hologic: HOLX
 The Home Depot: HD
 Honeywell: HON
 Hormel Foods: HRL
 Host Hotels & Resorts: HST
 Howmet Aerospace: HWM
 HP Inc.: HPQ
 Humana: HUM
 Huntington Bancshares: HBAN
 Huntington Ingalls Industries: HII
 IBM: IBM
 IDEXX Corporation: IEX
 Idexx Laboratories: IDXX
 Illinois Tool Works: ITW
 Illumina: ILMN
 Incyte: INCY
 Ingersoll Rand: IR
 Intel: INTC
 Intercontinental Exchange: ICE
 International Paper: IP

The Interpublic Group of Companies: IPG
 International Flavors & Fragrances: IFF
 Intuit: INTU
 Intuitive Surgical: ISRG
 Invesco: IVZ
 Invitation Homes: INRH
 IQVIA: IQV
 Iron Mountain: IRM
 J.B. Hunt: JBHT
 Jack Henry & Associates: JKHY
 Jacobs Solutions: J
 Johnson & Johnson: JNJ
 Johnson Controls: JCI
 JPMorgan Chase: JPM
 Juniper Networks: JNPR
 Kellogg's: K
 Keurig Dr Pepper: KDP
 KeyCorp: KEY
 Keysight: KEYS
 Kimberly-Clark: KMB
 Kimco Realty: KIM
 Kinder Morgan: KMI
 KLA Corporation: KLAC
 Kraft Heinz: KHC
 Kroger: KR
 L3Harris: LHX
 LabCorp: LH
 Lam Research: LRCX
 Lamb Weston: LW
 Las Vegas Sands: LVS
 Leidos: LDOS
 Lennar: LEN
 Lincoln Financial: LNC
 Linde plc: LIN
 Live Nation Entertainment: LYV
 LKQ Corporation: LKQ
 Lockheed Martin: LMT
 Loews Corporation: L
 Lowe's: LOW
 Lumen Technologies: LUMN
 LyondellBasell: LYB
 M&T Bank: MTB
 Marathon Oil: MRO
 Marathon Petroleum: MPC
 MarketAxess: MKTX
 Marriott International: MAR
 Marsh McLennan: MMC
 Martin Marietta Materials: MLM
 Masco: MAS
 Mastercard: MA

Match Group: MTCH
 McCormick & Company: MKC
 McDonald's: MCD
 McKesson: MCK
 Medtronic: MDT
 Merck & Co.: MRK
 Meta Platforms: META
 MetLife: MET
 Mettler Toledo: MTD
 MGM Resorts: MGM
 Microchip Technology: MCHP
 Micron Technology: MU
 Microsoft: MSFT
 Mid-America Apartment Communities: MAA
 Moderna: MRNA
 Mohawk Industries: MHK
 Molina Healthcare: MOH
 Molson Coors Beverage Company: TAP
 Mondelez International: MDLZ
 Monolithic Power Systems: MPWR
 Monster Beverage: MNST
 Moody's Corporation: MCO
 Morgan Stanley: MS
 The Mosaic Company: MOS
 Motorola Solutions: MSI
 MSCI: MSCI
 Nasdaq, Inc.: NDAQ
 NetApp: NTAP
 Netflix: NFLX
 Newell Brands: NWL
 Newmont: NEM
 News Corp (Class A): NWSA
 News Corp (Class B): NWS
 NextEra Energy: NEE
 Nike, Inc.: NKE
 NiSource: NI
 Nordson Corporation: NDSN
 Norfolk Southern Railway: NSC
 Northern Trust: NTRS
 Northrop Grumman: NOC
 Norwegian Cruise Line Holdings: NCLH
 NRG Energy: NRG
 Nucor: NUE
 Nvidia: NVDA
 NVR, Inc.: NVR
 NXP Semiconductors: NXPI
 O'Reilly Auto Parts: ORLY
 Occidental Petroleum: OXY
 Old Dominion: ODFL
 Omnicom Group: OMC

ON Semiconductor: ON
 ONEOK: OKE
 Oracle Corporation: ORCL
 Organon Co.: OGNN
 Otis Worldwide: OTIS
 Paccar: PCAR
 Packaging Corporation of America: PKG
 Paramount Global: PARA
 Parker Hannifin: PH
 Paychex: PAYX
 Paycom: PAYC
 PayPal: PYPL
 Pentair: PNR
 PepsiCo: PEP
 PerkinElmer: PKI
 Pfizer: PFE
 PG&E Corporation: PCG
 Philip Morris International: PM
 Phillips 66: PSX
 Pinnacle West: PNW
 Pioneer Natural Resources: PXD
 PNC Financial Services: PNC
 Pool Corporation: POOL
 PPG Industries: PPG
 PPL Corporation: PPL
 Principal Financial Group: PFG
 Procter & Gamble: PG
 Progressive Corporation: PGR
 Prologis: PLD
 Prudential Financial: PRU
 Public Service Enterprise Group: PEG
 PTC: PTC
 Public Storage: PSA
 PulteGroup: PHM
 Qorvo: QRVO
 Quanta Services: PWR
 Qualcomm: QCOM
 Quest Diagnostics: DGX
 Ralph Lauren Corporation: RL
 Raymond James: RJF
 Raytheon Technologies: RTX
 Realty Income: O
 Regency Centers: REG
 Regeneron: REGN
 Regions Financial Corporation: RF
 Republic Services: RSG
 ResMed: RMD
 Robert Half: RHI
 Rockwell Automation: ROK
 Rollins, Inc.: ROL

Roper Technologies: ROP
 Ross Stores: ROST
 Royal Caribbean Group: RCL
 S&P Global: SPGI
 Salesforce: CRM
 SBA Communications: SBAC
 Schlumberger: SLB
 Seagate Technology: STX
 Sealed Air: SEE
 Sempra Energy: SRE
 ServiceNow: NOW
 Sherwin-Williams: SHW
 Signature Bank: SBNY
 Simon Property Group: SPG
 Skyworks Solutions: SWKS
 The J.M. Smucker Company: SJM
 Snap-on: SNA
 SolarEdge: SEDG
 Southern Company: SO
 Southwest Airlines: LUV
 Stanley Black & Decker: SWK
 Starbucks: SBUX
 State Street Corporation: STT
 Steris: STE
 Stryker Corporation: SYK
 SVB Financial: SIVB
 Synchrony Financial: SYF
 Synopsys: SNPS
 Sysco: SYY
 T-Mobile US: TMUS
 T. Rowe Price: TROW
 Take-Two Interactive: TTWO
 Tapestry, Inc.: TPR
 Targa Resources: TRGP
 Target Corporation: TGT
 TE Connectivity: TEL
 Teledyne Technologies: TDY
 Teleflex: TFX
 Teradyne: TER
 Tesla, Inc.: TSLA
 Texas Instruments: TXN
 Textron: TXT
 Thermo Fisher Scientific: TMO
 TJX Companies: TJX
 Tractor Supply: TSCO
 Trane Technologies: TT
 TransDigm Group: TDG
 The Travelers Companies: TRV
 Trimble Inc.: TRMB
 Truist: TFC

Tyler Technologies: TYL
 Tyson Foods: TSN
 U.S. Bank: USB
 UDR, Inc.: UDR
 Ulta Beauty: ULTA
 Union Pacific Corporation: UNP
 United Airlines Holdings: UAL
 United Parcel Service: UPS
 United Rentals: URI
 UnitedHealth Group: UNH
 Universal Health Services: UHS
 Valero Energy: VLO
 Ventas: VTR
 Verisign: VRSN
 Verisk: VRSK
 Verizon: VZ
 Vertex Pharmaceuticals: VRTX
 VF Corporation: VFC
 Viatris: VTRS
 Vici Properties: VICI
 Visa Inc.: V
 Vornado Realty Trust: VNO
 Vulcan Materials Company: VMCC
 Wabtec: WAB
 Walgreens Boots Alliance: WBA
 Walmart: WMT
 Warner Bros. Discovery: WBD
 Waste Management: WM
 Waters Corporation: WAT
 WEC Energy Group: WEC
 Wells Fargo: WFC
 Welltower: WELL
 West Pharmaceutical Services: WST
 Western Digital: WDC
 WestRock: WRK
 Weyerhaeuser: WY
 Whirlpool Corporation: WHR
 Williams Companies: WMB
 Willis Towers Watson: WTW
 W. W. Grainger: GWG
 Wynn Resorts: WYN
 Xcel Energy: XEL
 Xylem Inc.: XYL
 Yum! Brands: YUM
 Zebra Technologies: ZBRA
 Zimmer Biomet: ZBH
 Zions Bancorporation: ZION
 Zoetis: ZTS

On selecting Choice 2, the user will get the following menu:

- 1.Pull the latest list of stocks from the S&P 500
- 2.Visualize any given stock
- 3.Get a personalized portfolio optimization

select a choice: 2

Please enter a ticker : AAPL

I.Daily Time Series

II.Weekly Time Series

III.Monthly Time Series

Please select a choice: 1

```
<class 'pandas.core.frame.DataFrame'>
```

Index: 100 entries, 99 to 0

Data columns (total 9 columns):

#	Column	Non-Null Count	Dtype
0	date	100	non-null
1	open	100	non-null
2	high	100	non-null
3	low	100	non-null
4	close	100	non-null
5	adjusted close	100	non-null
6	volume	100	non-null
7	7. dividend amount	100	non-null
8	8. split coefficient	100	non-null
	dtypes: datetime64[ns](1), float64(4), int64(1), object(3)		
	memory usage: 7.8+ KB		

In this menu, if Choice 1 is selected, the user will be asked to input a stock of their choice, of which a stock price v/s time series graph will be displayed.

In this case the user decided to see the graph of the ticker “AAPL” which, on close inspection of the S&P 500, tells us, belongs to the company Apple, whose chart is displayed.

DAILY - AAPL

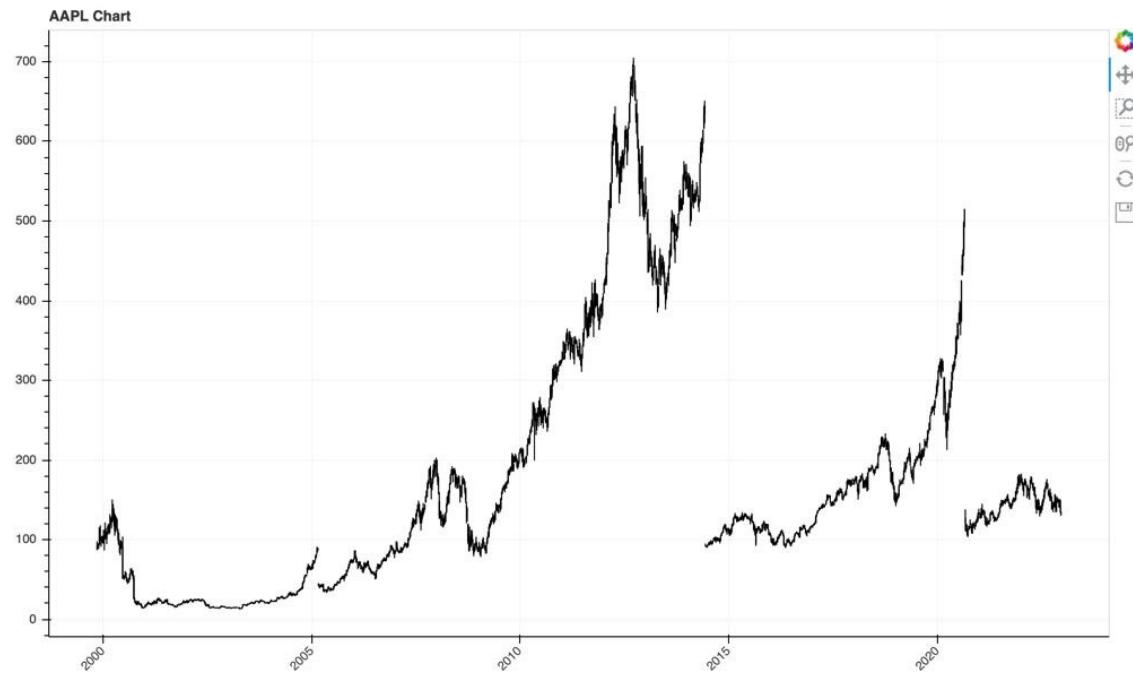


If the user would have selected choice 2, then a monthly graph would be displayed.

MONTHLY - AAPL

```
Please enter a ticker : AAPL
I.Daily Time Series
II.Weekly Time Series
III.Monthly Time Series

Please select a choice: 2
<class 'pandas.core.frame.DataFrame'>
Index: 1207 entries, 1206 to 0
Data columns (total 6 columns):
 #   Column   Non-Null Count Dtype  
 0   date     1207 non-null    datetime64[ns]
 1   open     1207 non-null    float64  
 2   high     1207 non-null    float64  
 3   low      1207 non-null    float64  
 4   close    1207 non-null    float64  
 5   volume   1207 non-null    int64   
dtypes: datetime64[ns](1), float64(4), int64(1)
memory usage: 66.0+ KB
```

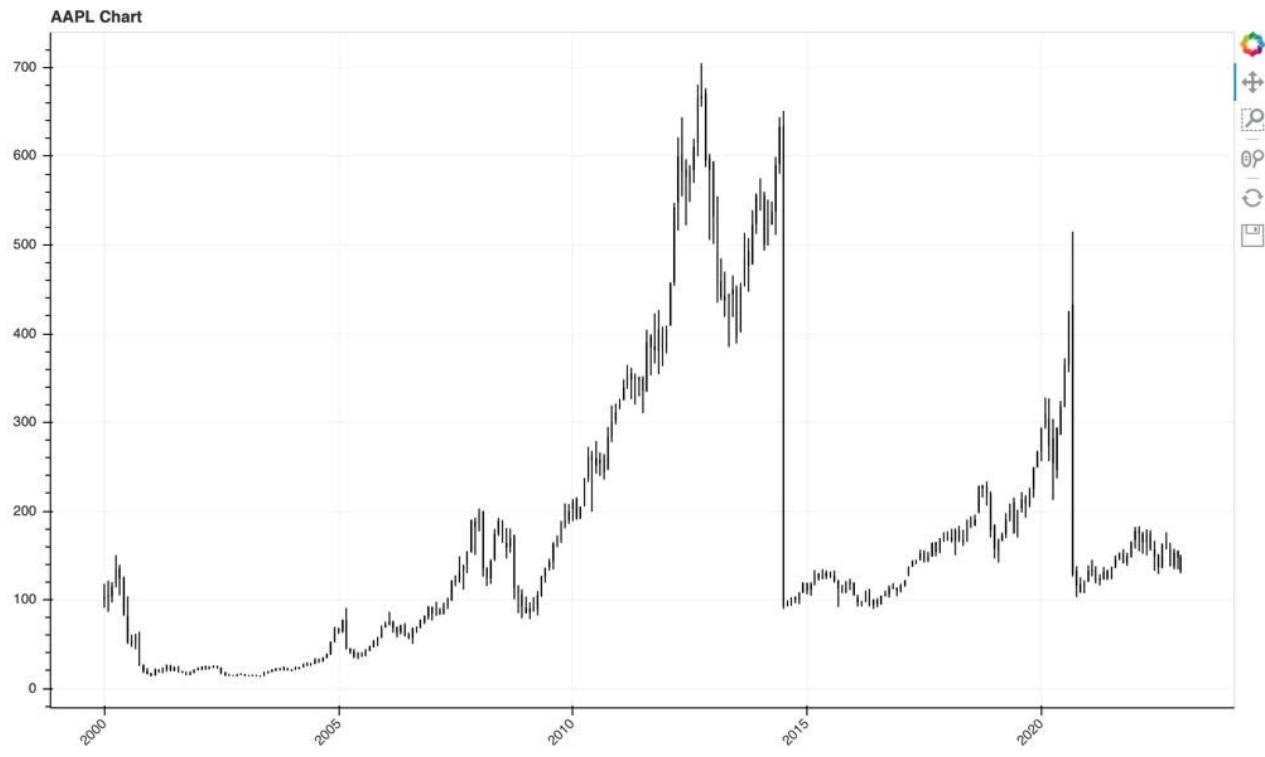


If the user would have selected choice 3, then a yearly graph would be displayed.

```
Please enter a ticker : AAPL
I.Daily Time Series
II.Weekly Time Series
III.Monthly Time Series

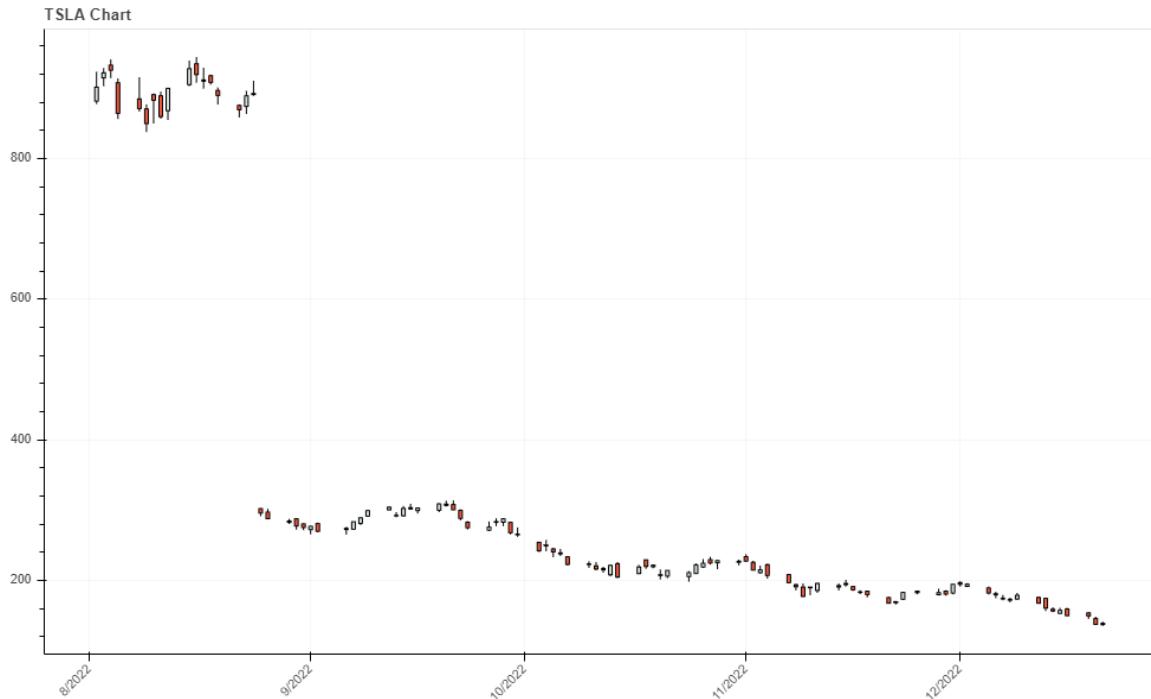
Please select a choice: 3
<class 'pandas.core.frame.DataFrame'>
Index: 277 entries, 276 to 0
Data columns (total 6 columns):
 #   Column   Non-Null Count   Dtype   
 0   date     277 non-null    datetime64[ns]
 1   open     277 non-null    float64  
 2   high     277 non-null    float64  
 3   low      277 non-null    float64  
 4   close    277 non-null    float64  
 5   volume   277 non-null    int64    
dtypes: datetime64[ns](1), float64(4), int64(1)
memory usage: 15.1+ KB
```

YEARLY - AAPL

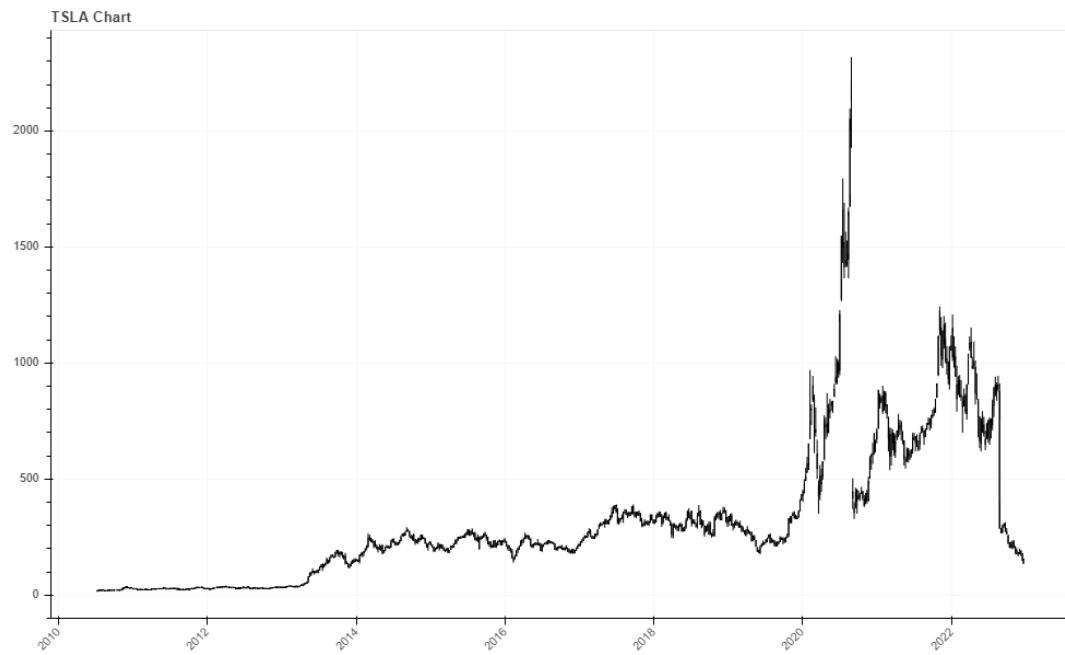


Some Examples for Understanding

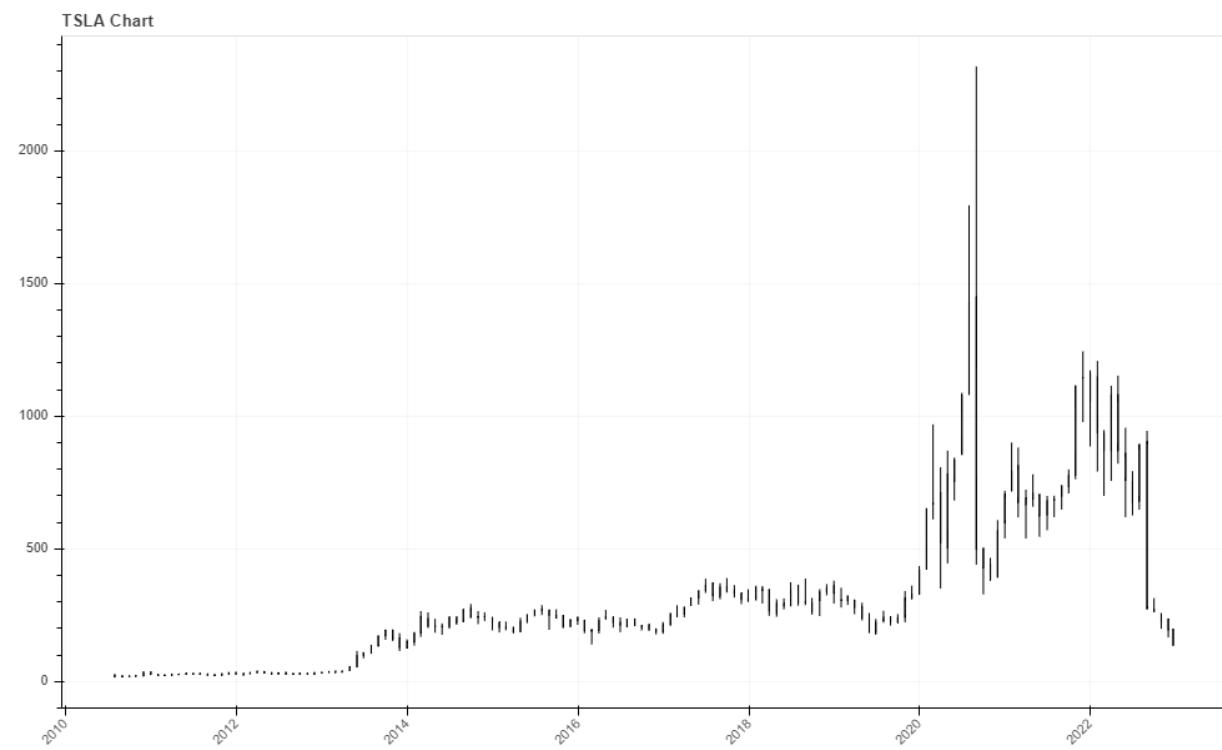
Daily - Tsla



WEEKLY - TSLA



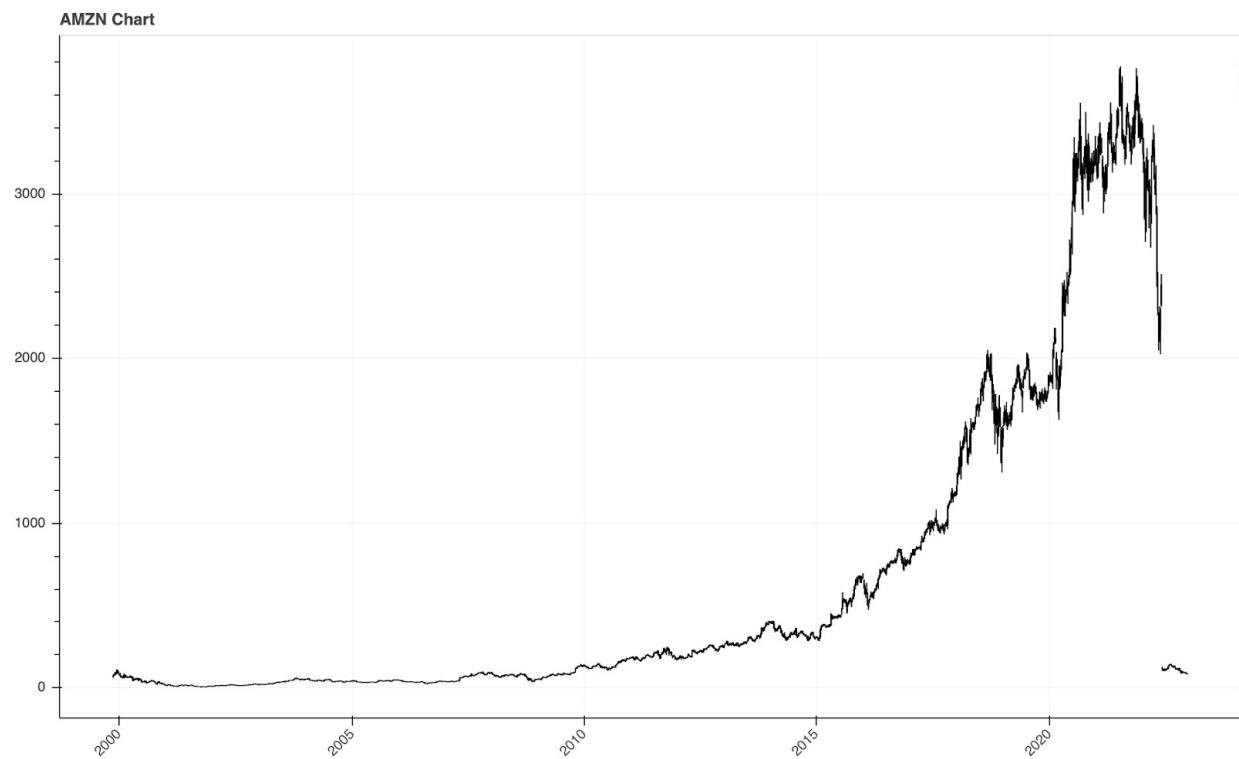
MONTHLY - TSLA



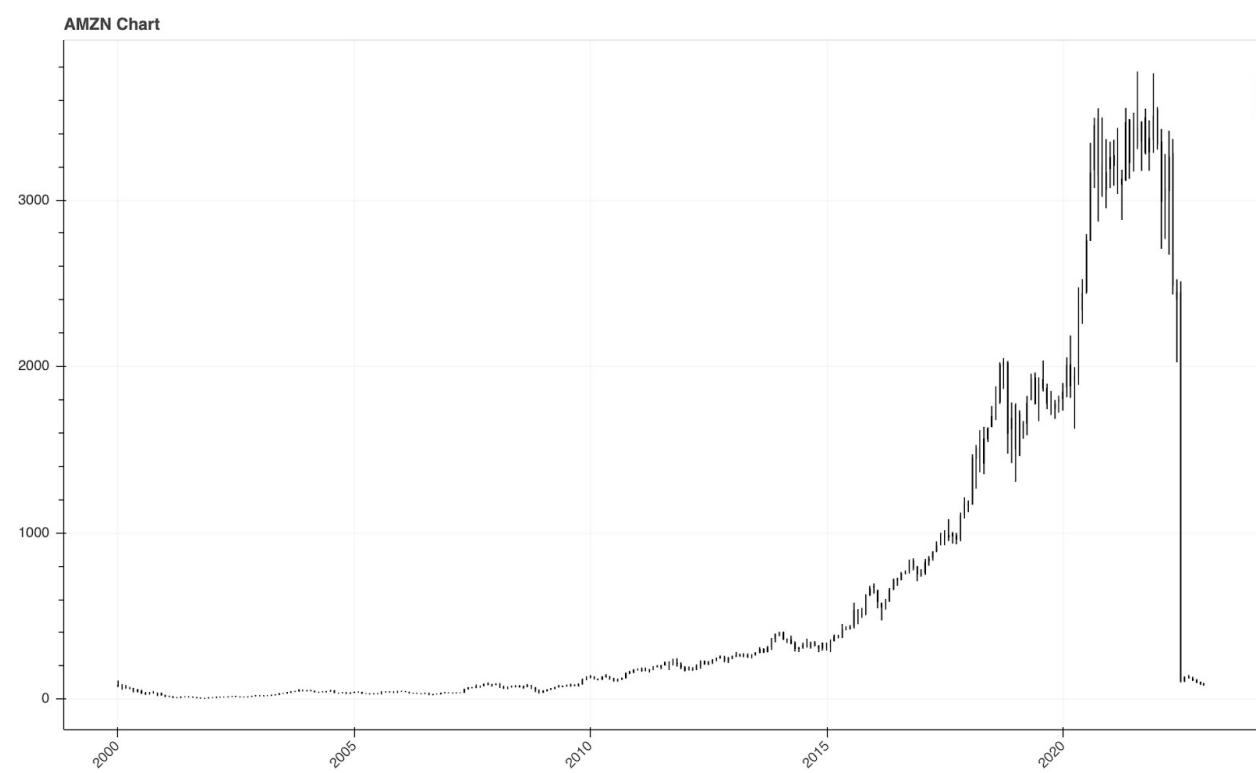
DAILY - AMZN



WEEKLY - AMZN



MONTHLY - AMZN



In the main menu, if the user selects choice 3, then they would get the option to enter upto 10 stock tickers, of which, a markowitz portfolio optimization would be ran, and the portfolio with the highest sharp ratio would be computed and displayed to the user. This result is in accordance with the Modern Portfolio, as referenced in the Introduction.

- ```
1.Pull the latest list of stocks from the S&P 500
2.Visualize any given stock
3.Get a personalized portfolio optimization
```

```
select a choice: 3
```

```
Please enter a list of 10 stock tickers: {'MRNA':0.48875,'PFE':0.0,'JNJ':0.0,'GOOGL':0.29459,'FB':0.0,'AAPL':0.16967,'COST':0.0,'WMT':0.0,'KR':0.04698,'JPM':0.0,'BAC':0.0,'HSBC':0.0}
```

```
{'MRNA': 0.48875, 'PFE': 0.0, 'JNJ': 0.0, 'GOOGL': 0.29459, 'FB': 0.0, 'AAPL': 0.16967,
'COST': 0.0, 'WMT': 0.0, 'KR': 0.04698, 'JPM': 0.0, 'BAC': 0.0, 'HSBC': 0.0}
```

**Expected annual return: 225.7%**

**Annual volatility: 44.6%**

**Sharpe Ratio: 5.02**

**Expected annual return: 24.5%**

**Annual volatility: 20.0%**

**Sharpe Ratio: 1.12**

**Discrete allocation: {'MRNA': 112, 'GOOGL': 10, 'AAPL': 113, 'KR': 114}**

**Funds remaining: \$928.79**