

Kingdom of Saudi Arabia  
Ministry of Education  
Qassim University  
College of Computer  
Information Technology Dept.



المملكة العربية السعودية  
وزارة التعليم  
جامعة القصيم  
كلية الحاسب  
قسم تقنية المعلومات

# Real Time Stock Market Simple Visualization

*Programmed by:*

**Sara Alrumih**

**Leena Alhajjaj**

*Instructor:*

**Dr. Amal Alshargabi**

## Table of Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Introduction . . . . .	2
1.2	Project Idea and Scope . . . . .	2
1.3	Project Objectives . . . . .	2
1.4	Tools used . . . . .	2
<b>2</b>	<b>PROGRAM OVERVIEW</b>	<b>3</b>
2.1	Program Overview . . . . .	4
2.2	Program Guide . . . . .	4
2.3	Program Code . . . . .	7
2.4	Challenges and Limitations . . . . .	8
2.5	Future Work . . . . .	8
2.6	Conclusion . . . . .	8

## Chapter 1

# INTRODUCTION

# INTRODUCTION

## 1.1 Introduction

As a requirement of big data analysis (IT474) subject, we developed this program that satisfies the subject's project requirement which is: real time big data visualization. The main purpose of this program is to provide a simple visualisation of stock market prices for interested.

## 1.2 Project Idea and Scope

This project discusses the development of a real time data visualization program of stock market prices. People interested in financial and stock market can benefit from this program, as it provides a simple stock prices visualization to them. This project supports making financial decisions from visualizing the scatter graph.

## 1.3 Project Objectives

This project aims to visual real time stock market prices in a simple way. By presenting one scatter graph that shows the current prices to a specific stock market chose by the user. The graph is timely updated, i.e. every second, minute, or hourly.

## 1.4 Tools used

The program developed using Python programming language, and the following Python packages:

- matplotlib.pyplot  
A state-based interface to matplotlib. It provides a MATLAB-like way of plotting. pyplot is mainly intended making interactive plots.
- requests  
Python package that allow sending HTTP requests.
- json  
Python package used to work with JSON data.
- time  
Python package that provides various time-related functions.
- tkinter  
The standard Python interface to the Tk graphical user interface (GUI) toolkit.

Moreover, market prices were loaded from (Cryptocompare.com) website. Cryptocompare.com is a global cryptocurrency market data provider, giving institutional and retail investors access to real-time, high-quality, and reliable market. CryptoCompare provides a comprehensive, holistic overview of the market.

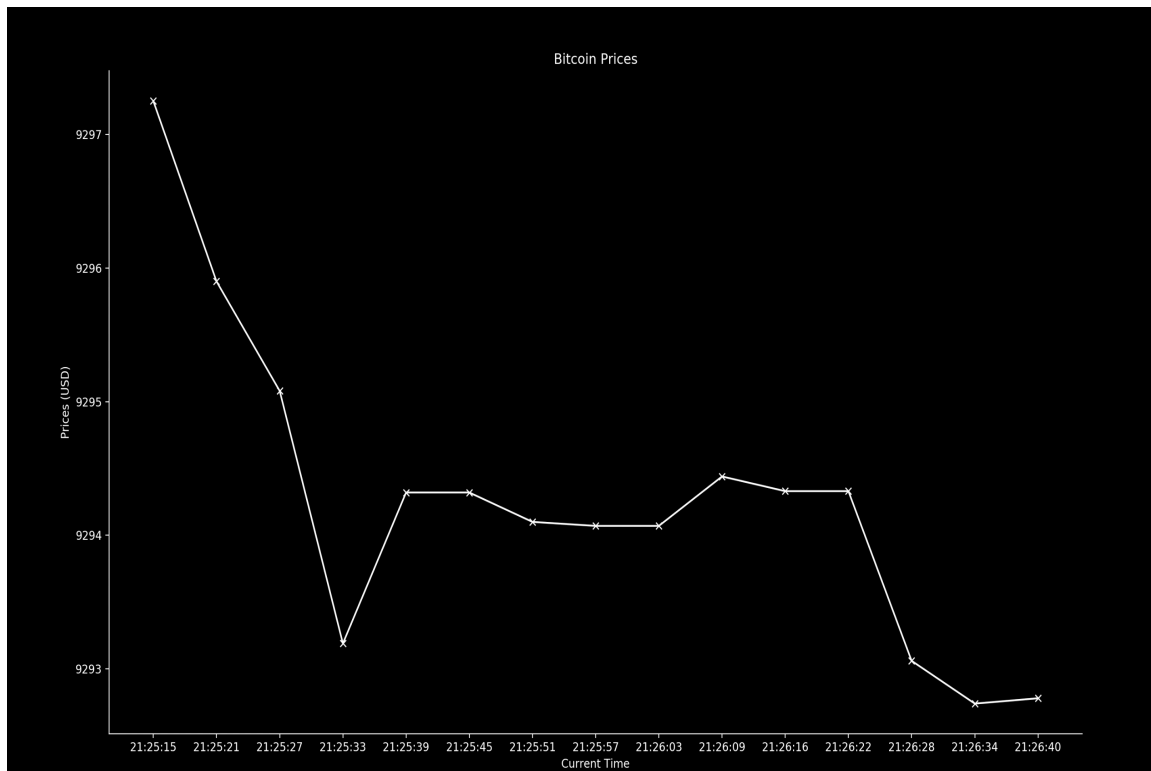
## Chapter 2

# PROGRAM OVERVIEW

# PROGRAM OVERVIEW

## 2.1 Program Overview

As all stock market visualization programs and websites we have seen are a little bit complicated and shows various information, we developed a simple program that present a real-time scatter graph of a specific stock market prices. Scatter graph is a graph of plotted points that shows the relationship between two sets of data. In this project, the scatter graph shows the relationship of stock market prices in the current time, the presented plots are connected via a line to simplify the presentation. The presented scatter graph has a dark mode, i.e. black background with white axis-es values and plots. In order to use and benefit from this program, users will be given the program code to run it in an environment with python 3 support, as well as a guide of how to use and configure this program. Before running the program users need to configure it by specifying the stock symbol i.e. an abbreviation used to uniquely identify stock market e.g. BTC for bitcoin, time to update the graph, and the graph title as detailed in section 2 (Program Guide). The following figure represent a screen shot from bitcoin graph updated each 5 seconds.



## 2.2 Program Guide

The following steps are necessary in order to use and benefit from this program.

1. Step 1: Setting the environment

In this step a python 3 supporting environment is needed. You can install python 3 or update your python version from python website: <https://www.python.org>. After setting the environment you need to create python file using your command line i.e. terminal or shell, or any graphical user interface environment that support python e.g. PyCharm,

name it e.g. stock.py than paste the code in it. Here is an example of creating python file in command line:

1. Open command line (terminal), then go to Desktop (optional)

```
(base) Saras-MacBook-Pro-2:~ sara$ cd Desktop
```

2. Create python file. You can type nano filename.py to create python file as in the example bellow.

```
(base) Saras-MacBook-Pro-2:Desktop sara$ nano stock.py
```

3. Copy the code in the program code section and paste it in the python file you have created.

```
GNU nano 2.0.6 File: stock.py

import matplotlib.pyplot as plt
import requests, json, time
import tkinter as tk
from tkinter import messagebox

# define variables for data retrieval
times = [] #stores the values of the time when we retrieve price data in a list.
currency = "BTC" # holds the stock symbol we will be fetching data for
prices = [] #dictionary that holds the list of prices for currency defined.
plt.style.use('dark_background')

def retrieve_data():

    # append new time to list of times
    times.append(time.strftime('%H:%M:%S'))

    # make request to API and get response as object
    api_url = "https://min-api.cryptocompare.com/data/pricemulti?fsyms={}&tsyms=USD".format(currency)
    try:
        response = json.loads(requests.get(api_url).content)

        # append new price to list of prices for graph
        price = response[currency].get('USD')
        prices.append(price)
    except:
        root = tk.Tk()
        root.withdraw()
        messagebox.showerror("Oops", "The stock market is close right now!\nTry the system later, Thank you.")
        exit(0)

    # clear plot
    plt.clf()

    # x axis values
    x = times

    # corresponding y axis values
    y = prices

    # avoid overlapping in x axis
    if len(x)>15:
        del x[0]
        del y[0]

    # plotting the points
    plt.plot(x,y,marker="x",color="w")

    # naming the x axis
    plt.xlabel('Current Time')

    # naming the y axis
    plt.ylabel('Prices (USD)')

    # remove top and right borders
    plt.gca().spines['top'].set_visible(False)
    plt.gca().spines['right'].set_visible(False)

    # giving a title to my graph
    plt.title('Bitcoin Prices')

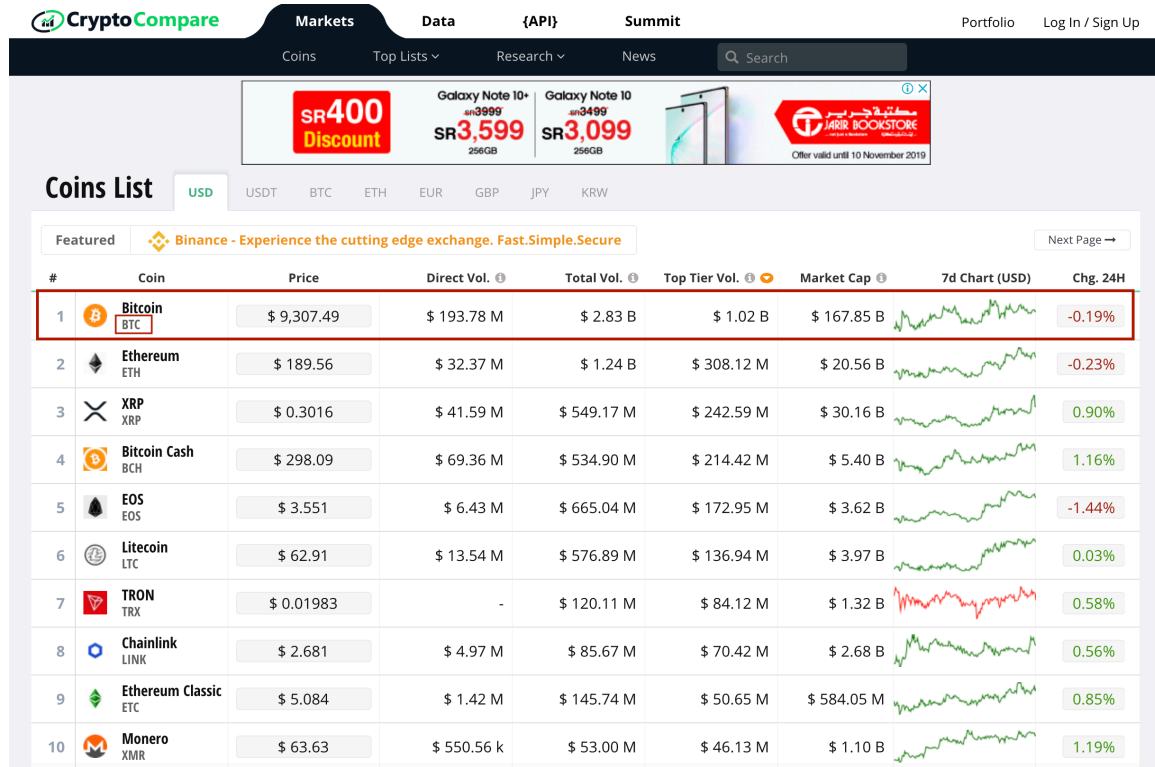
    # update every 5 seconds
    plt.pause(5)

while(True):
    retrieve_data()
```

## 2. Step 2: Configuration

You need to determine three elements: stock symbol, update period, and graph title. To specify the stock symbol:

1. Search in cryptocompare website (<https://www.cryptocompare.com>) for the stock symbol of your stock market.



2. Write it in line 8 as shown in the figure bellow. By default BTC for bitcoin will be written, overwrite it to the desired stock symbol.

```
currency = "BTC"
```

To specify update period, make changes in this line (line 62). Keep in mind to convert it to seconds, here is a website that will help you in conversion: <http://www.easysurf.cc/utime.htm>. By default the graph will be updated every 5 seconds.

```
plt.pause(5)
```

To update the graph title, which will be by default "Bitcoin prices", overwrite your desired title in this line (line 59).

```
plt.title('Bitcoin Prices')
```

If you are interested in editing the time format from (hour : minutes : seconds) to another format, make changes in line 15 as described bellow:  
(Hour : minutes) => times.append(time.strftime('%H:%M'))



```
(Hour) => times.append(time.strftime('%H'))  
(month - day) => times.append(time.strftime('%m-%d'))
```

### 3. Step 3: Running

After you finished with the configuration it is the time to run the program. To run it type `python3 filename.py` in the command line, in our example `python3 stock.py`. The program will run immediately.

```
(base) Saras-MacBook-Pro-2:Desktop sara$ python3 stock.py
```

For any confusions or questions feel free to contact us via our email, [saraalrumih@gmail.com](mailto:saraalrumih@gmail.com).

## 2.3 Program Code

```
1 import matplotlib.pyplot as plt  
2 import requests, json, time  
3 import tkinter as tk  
4 from tkinter import messagebox  
5  
6 # define variables for data retrieval  
7 times = [] #stores the values of the time when we retrieve price data in a list.  
8 currency = "BTC" # holds currency stock symbol we will be fetching data for  
9 prices = [] #dictionary that holds the list of prices for currency defined.  
10 plt.style.use('dark_background')  
11  
12 def retrieve_data():  
13  
14     # append new time to list of times  
15     times.append(time.strftime('%H:%M:%S'))  
16  
17     # make request to API and get response as object  
18     api_url = "https://min-api.cryptocompare.com/data/pricemulti?fsyms={}&tsyms=USD  
19     ".format(currency)  
20     try:  
21         response = json.loads(requests.get(api_url).content)  
22  
23         # append new price to list of prices for graph  
24         price = response[currency].get('USD')  
25         prices.append(price)  
26     except:  
27         root = tk.Tk()  
28         root.withdraw()  
29         messagebox.showerror("Oops", "The stock market is close right now!\nTry the  
30         system later, Thank you.")  
31         exit(0)  
32  
33     # clear plot  
34     plt.clf()  
35  
36     # x axis values  
37     x = times  
38  
39     # corresponding y axis values  
40     y = prices  
41  
42     # avoid overlapping in x axis
```

```
41     if len(x)>15:
42         del x[0]
43         del y[0]
44
45     # plotting the points
46     plt.plot(x,y,marker="x",color="w")
47
48     # naming the x axis
49     plt.xlabel('Current Time')
50
51     # naming the y axis
52     plt.ylabel('Prices (USD)')
53
54     # remove top and right borders
55     plt.gca().spines['top'].set_visible(False)
56     plt.gca().spines['right'].set_visible(False)
57
58     # giving a title to my graph
59     plt.title('Bitcoin Prices')
60
61     # update graph every 5 seconds
62     plt.pause(5)
63
64 while(True):
65     retrieve_data()
```

## 2.4 Challenges and Limitations

This program represent the simplest form of stock prices visualizing. We tried to make the configuration done through GUI but the pyplot window does not support adding widgets to it. Moreover, we tried to use another python package for graphical representation. It did allow us to make GUI configuration control, but it did not support the timely updating feature.

## 2.5 Future Work

The current work present a simple graph of real time stock prices. However, it require some programming effort by the user. We will continue our work in the future to eliminate the programming effort from users by transforming the configuration to GUI control.

## 2.6 Conclusion

In conclusion, we had fun challenging time developing this program and we believe it will get economisers and financiers attention due to its simplicity. We will also work on improving it.