Alvin Radoncic

Professor Kevin Ryan

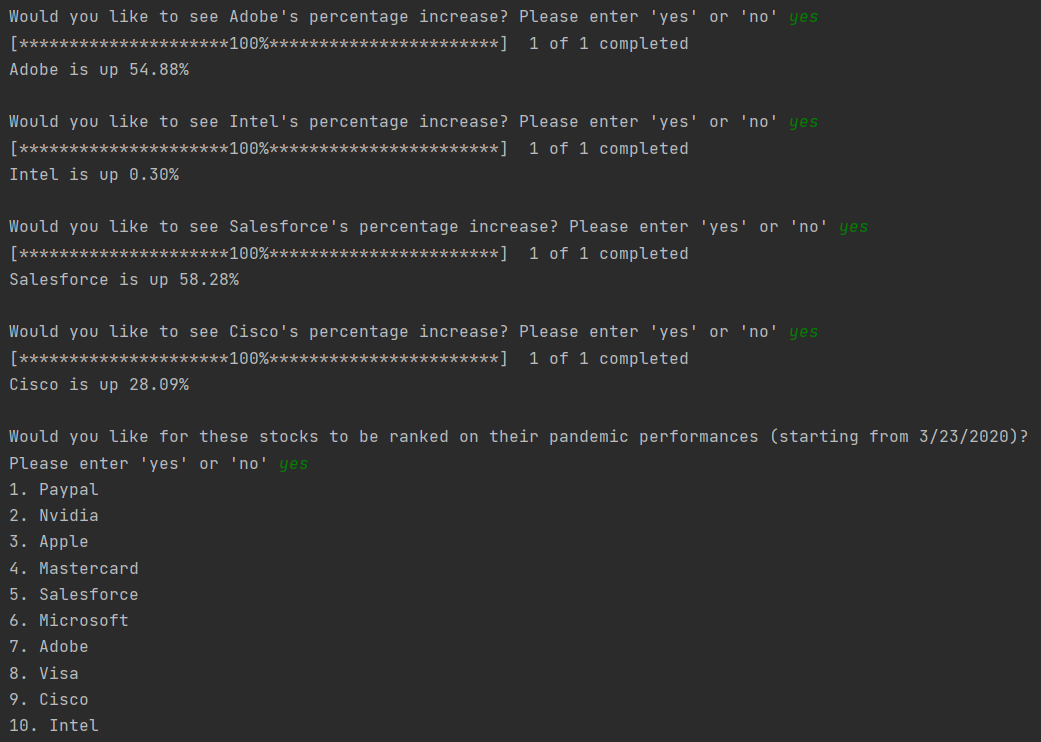
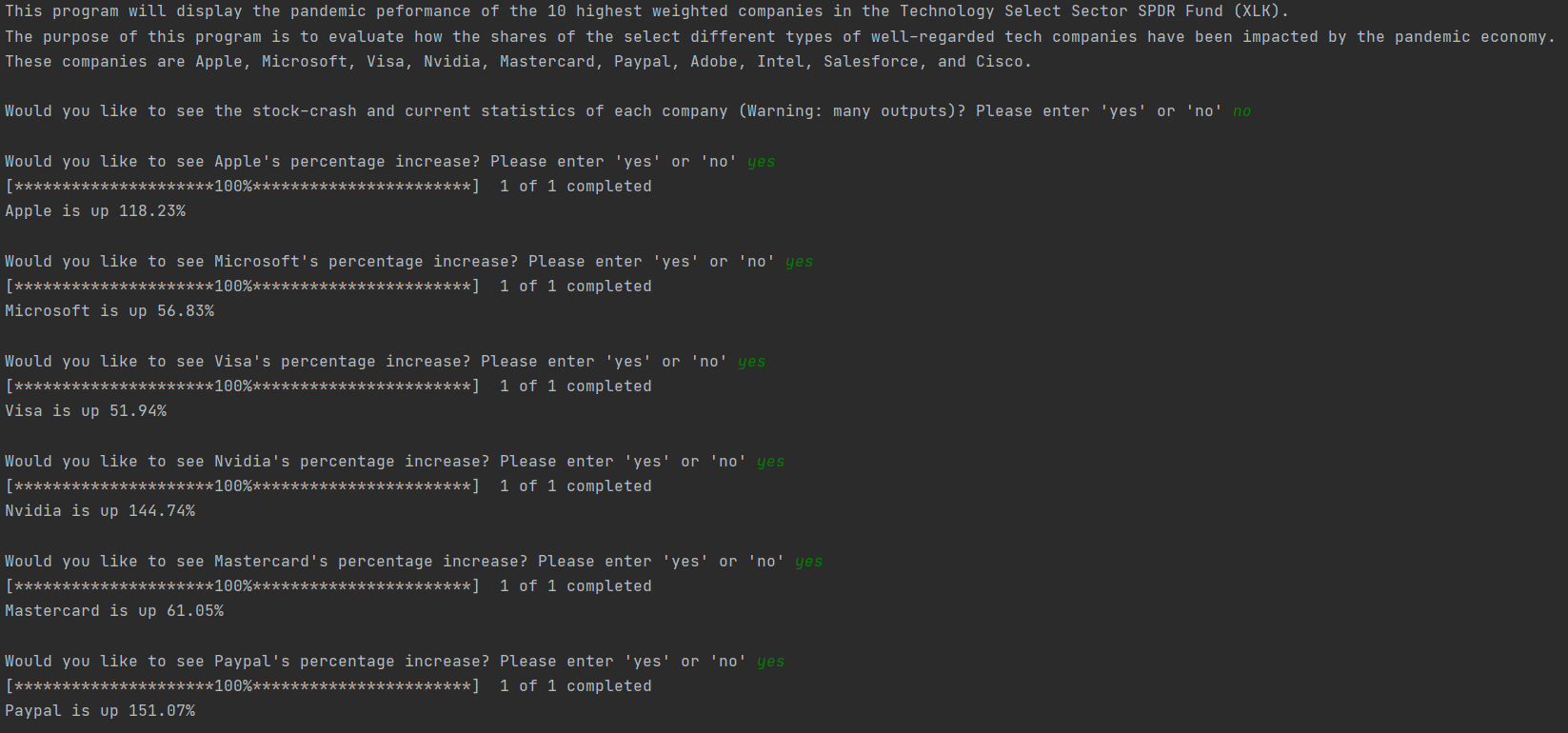
CS-110-A

December 12th, 2020

Final Project Report

**Section One; “Overview and Summary of Project”**

What my program achieves is collecting data from finance.yahoo.com, restricting the dataframes’ “Date” columns to accurately track data from the pandemic-driven stock market crash on March 23rd of this year to today (the program goes by December 12th), calculates the percentage increase of the ten highest weighted tech stocks in the Technology Select Sector SPDR Fund (XLK), and ranks them. I chose the ten stocks because they contained a variety of different kinds of major tech companies while also allowing for comparative review. These companies are Apple, Microsoft, Visa, Nvidia, Mastercard, Paypal, Adobe, Intel, Salesforce, and Cisco. Due to my beginner knowledge of programming, the only interface is the console, for we have just learned very simple graphical user interfaces. Additionally, since my project mainly grabs data from 10 specific data frames and makes calculations off it, the only user inputs I can do are under if-else statements. The user is asked simple “yes or no” options on whether they want to see parts of the dataframes, the percentage increase of each company, and how they are ranked based on their percentage increases. Here is a sample of what a user would see:



**Section Two; “Target Audience”**

The target audience would definitely be retail investors who always tend to seek for information on share prices on the internet. This could also help business students who have projects and studies to conduct, and this program would definitely give them useful information to work with. My program, as I expected when I had begun to build it, deals simple information to the user, so I would assume more advanced people in the world of business either have other tools to refer to or have built their very own.

**Section Three; “Specific Programming Techniques Used”**

The structure of the program is something that I would really want to improve in the near future. I start off by importing the yfinance library and the pandas library. Using yfinance helped me address the possible problems with file inputs, as it allows the code to download data directly from Yahoo Finance using its download class and inputting the necessary dates as the date range meant to be imported, as well as a list of a tickers so that the program knows what specific stock data it should be taking. Pandas allowed me to make the data look more organized, as well as it allowed me to grab specific values from each stock’s dataframe that would be used for later calculations. I then created a function called data\_download(), which has a list of all 10 tickers, and uses a for loop that is able to download data of each ticker and print it out as two pieces: the data of March 23rd (the market crash) and its current data.

Next, I created ten separate functions, each tracking a specific stock’s (out of the 10 being observed) percentage gain since the pandemic caused the stock market to crash. In each function, the stock’s data is downloaded, then the iloc function is used to extract the closing prices at March 23rd and today, which are each made into a variable. Lastly, these two variables are used to formulate the percentage gain, which is then assigned to a variable called “performance”, which is the return value of each of these ten functions.

I tried to create a function called performance-ranking() that would automatically rank each stock by their “performance”, but my attempts of using a double for-loop failed, and I realized that I needed to learn how to use dictionary functions to do so, so this was an overall shortcoming of the program.

Everything that is outputted to the user is within the final main() function. I have 12 variables in it. The first one is Data-Option, which asks the user whether or not they want to see the data of all 10 stocks from the 2 important dates. If they say yes, the dataframes are shown, and any other answer passes on to the next variable. The next 10 variables act similarly, but regard each of the 10 stocks, asking the user if they want to see their percentage gain. The last variable is Ranking\_Option, which acts the same as well, but will output the ranking of the best performing stocks of the pandemic, based on the values acquired from the 10 functions of each stock. This main() function is the culmination of decision structures, variable assignments, values returned from previous functions, and many pieces of data. I would also like to note that each output, which all occurs in main(), uses .format.

**Section Four; “Challenge(s)”**

The major challenge I had gone through was when I realized that I was unable to allow the program itself to rank the stocks on their performances. This really hindered the quality of the program and any future modularity it could have. I was struggling with finding a seamless way of importing lots of data until I found out about the use of the yfinance library. The same went for dealing with the data frames that were produced by yfinance, and the pandas library functions, especially iloc, allowed me to be able to take specific values from data frames. I was also trying to use my own crafted classes and objects, but I ended up realizing that I’m not as confident as I want to be at OOP, so I ended up tracking the performance of each stock by using 10 individual functions.

**Section Five; “Future Extensions”**

I would like to optimize the program before making add-ons. This would primarily regard the program being able to have a working function ranking each stock’s performance and having only one part of it that takes the task of tracking all stock performances. Optimization is very important and I do not feel I have done the best that I could.

As for future add-ons, the first thing that crosses my mind is having the user be able to input any valid ticker, which would then undergo the processes that the program would have to offer. This is very important for the sake of giving the user a far more modular experience. I would also have the user be able to request a chart be made of the stock performances they seek to observe, and perhaps give them the ability to adjust the date range.