**Final Portfolio** for

Python Programming – ICT 4370

Master of Science, Information & Communication Technology

Mauricio Beltran

University of Denver University College

Date June 4, 2023

Faculty: Abeer A. Khaleq, PhD

Director: Cathie Wilson, MS

Dean: Michael J. McGuire, MLS

The following is a breakdown of the additional functionality that I have added to the code, which week have been working on during the course.

Cumulative returns measure the total change in the value of an investment over a specific period, taking into account both the price appreciation and any dividends or interest received. It shows the overall performance of the investment over time, considering all the gains and losses.

In this code, calculating cumulative returns is important because it allows us to assess the overall performance of the portfolio stocks. By plotting the cumulative returns, we can visually analyze the growth or decline of the investment value over time. It provides insights into the long-term profitability or loss of the portfolio.

By examining the cumulative returns, investors can evaluate the effectiveness of their investment strategies, compare the performance of different stocks, and make informed decisions about their portfolio allocation. It helps assess the profitability and risk associated with the investment portfolio and aids in determining whether the investment strategy is achieving the desired financial goals.

Reflecting on the Portfolio Assignment, several areas could benefit from improvement. One such area is error handling. Implementing more robust error-handling mechanisms throughout the code would enhance its reliability and make it more user-friendly. By incorporating proper exception handling, the code can gracefully handle errors and provide meaningful error messages to users, helping them understand and resolve issues more efficiently.

In response to instructor feedback, several updates were made to the code. One notable addition was incorporating a database to store portfolio and stock data. This allowed for more efficient data storage and retrieval, enabling the code to handle larger datasets. The code was also modified to read data from CSV and JSON files, making it more flexible regarding data sources. Additionally, the code was enhanced to generate portfolio reports, providing users with a comprehensive overview of their investments. The ability to plot portfolio values over time was also added, enabling users to visualize the performance of their investments.

While these updates addressed the feedback received, there are still areas where further improvement can be made. For instance, the code could benefit from more advanced data analysis techniques and additional visualization libraries. By incorporating more sophisticated data analysis methods, such as statistical modeling or correlation analysis, deeper insights into portfolio performance can be gained. Exploring additional visualization libraries beyond Matplotlib, such as Plotly or Seaborn, could also enhance the visual representation of data and improve the overall user experience.

In conclusion, the Portfolio Assignment underwent significant improvements based on instructor feedback. However, there is still room for enhancement in error handling, modularity, advanced data analysis, visualization techniques, integration of external APIs, and documentation accessibility. By continuing to refine these aspects, the code can become more robust, flexible, and user-friendly, providing a more comprehensive and valuable experience for portfolio analysis.