Task 1: Algorithmic Trading Adventure

Scenario: Alex, a budding programmer and finance enthusiast, embarks on an algorithmic trading adventure with a budget of $5000. Their mission? To develop a tool that leverages Python to make informed decisions in the stock market, using a class-based approach for flexibility.

Task Steps:

Initializing Class: Create a class to encapsulate the trading strategy, allowing initialization with parameters such as symbol, from date, and to date. Example:
class_Name("AAPL","2018-01-01","2023-12-31")

Setting the Stage: Install the yfinance library to access historical market data.

Data Acquisition: Download historical data for the specified symbol within the provided date range.

Data Cleanup: Filter out duplicate data points and handle NaN values by forward filling.

Analytical Insights: Compute the moving averages for 50 and 200 days.

Golden Opportunity: Identify the golden cross, signaling a bullish trend, and take a buying position.

Investment Strategy: Determine the maximum quantity of shares to purchase within the $5000 budget.

Timely Actions: When the golden cross reverses, sell the position and close the trade. When you are in a position you can't buy other stock.

Final Touches: Forcefully close the position on the last row if a position is still open.

Evaluation: Calculate profits or losses to assess the success of the trading strategy.

Through this adventure, Alex not only hones their programming skills but also gains valuable insights into the dynamic world of finance, all while managing their investments wisely.

My Approach:

Initially , I experimented on the .ipynb (python notebook) to execute and perform the steps cell by cell and when I finished the entire thing , I kept this raw notebook as it is and started converting each steps into a method to use it in the class and later wrote the .py file .
Added some visualization to the .py file for better readability .