Codealpha Task 2

import yfinance as yf

class StockPortfolio:

def \_\_init\_\_(self):

self.portfolio = {}

def add\_stock(self, ticker, shares):

"""Add a new stock to the portfolio."""

try:

stock\_data = yf.Ticker(ticker)

stock\_info = stock\_data.info

stock\_price = stock\_data.history().iloc[-1]['Close']

if ticker in self.portfolio:

self.portfolio[ticker]['shares'] += shares

else:

self.portfolio[ticker] = {

'name': stock\_info['shortName'],

'shares': shares,

'price': stock\_price,

'total\_cost': shares \* stock\_price

}

print(f"{shares} shares of {stock\_info['shortName']} ({ticker}) added to the portfolio.")

except Exception as e:

print(f"Error adding {ticker}: {e}")

def remove\_stock(self, ticker, shares):

"""Remove shares of a stock from the portfolio."""

if ticker not in self.portfolio:

print(f"{ticker} is not in the portfolio.")

return

if shares > self.portfolio[ticker]['shares']:

print(f"You don't have enough shares of {ticker} to sell.")

return

self.portfolio[ticker]['shares'] -= shares

self.portfolio[ticker]['total\_cost'] -= shares \* self.portfolio[ticker]['price']

if self.portfolio[ticker]['shares'] == 0:

del self.portfolio[ticker]

print(f"{shares} shares of {ticker} removed from the portfolio.")

else:

print(f"{shares} shares of {ticker} removed from the portfolio.")

def get\_portfolio\_value(self):

"""Calculate the total value of the portfolio."""

total\_value = 0

for ticker, stock\_info in self.portfolio.items():

try:

stock\_data = yf.Ticker(ticker)

current\_price = stock\_data.history().iloc[-1]['Close']

stock\_value = stock\_info['shares'] \* current\_price

total\_value += stock\_value

except Exception as e:

print(f"Error fetching data for {ticker}: {e}")

return total\_value

def print\_portfolio(self):

"""Print the current state of the portfolio."""

print("Your Portfolio:")

print("----------------")

for ticker, stock\_info in self.portfolio.items():

try:

stock\_data = yf.Ticker(ticker)

current\_price = stock\_data.history().iloc[-1]['Close']

stock\_value = stock\_info['shares'] \* current\_price

print(f"{stock\_info['name']} ({ticker}) - Shares: {stock\_info['shares']}, Current Price: ${current\_price:.2f}, Value: ${stock\_value:.2f}")

except Exception as e:

print(f"Error fetching data for {ticker}: {e}")

portfolio\_value = self.get\_portfolio\_value()

print(f"\nTotal Portfolio Value: ${portfolio\_value:.2f}")

if \_\_name\_\_ == "\_\_main\_\_":

portfolio = StockPortfolio()

while True:

action = input("Enter 'add', 'remove', 'view', or 'exit': ").lower()

if action == 'add':

ticker = input("Enter the stock ticker: ").upper()

shares = int(input("Enter the number of shares: "))

portfolio.add\_stock(ticker, shares)

elif action == 'remove':

ticker = input("Enter the stock ticker: ").upper()

shares = int(input("Enter the number of shares to sell: "))

portfolio.remove\_stock(ticker, shares)

elif action == 'view':

portfolio.print\_portfolio()

elif action == 'exit':

break

else:

print("Invalid action. Please try again.")

Output:

Enter 'add', 'remove', 'view', or 'exit': add

Enter the stock ticker: AAPL

Enter the number of shares: 100

100 shares of Apple Inc. (AAPL) added to the portfolio.

Enter 'add', 'remove', 'view', or 'exit': add

Enter the stock ticker: GOOGL

Enter the number of shares: 50

50 shares of Alphabet Inc. (GOOGL) added to the portfolio.

Enter 'add', 'remove', 'view', or 'exit': view

Your Portfolio:

Apple Inc. (AAPL) - Shares: 100, Current Price: $173.45, Value: $17345.00

Alphabet Inc. (GOOGL) - Shares: 50, Current Price: $2345.67, Value: $117283.50

Total Portfolio Value: $134628.50

Enter 'add', 'remove', 'view', or 'exit': remove

Enter the stock ticker: AAPL

Enter the number of shares to sell: 25

25 shares of AAPL removed from the portfolio.

Enter 'add', 'remove', 'view', or 'exit': view

Your Portfolio:

Apple Inc. (AAPL) - Shares: 75, Current Price: $173.45, Value: $13008.75

Alphabet Inc. (GOOGL) - Shares: 50, Current Price: $2345.67, Value: $117283.50

Total Portfolio Value: $130292.25

Enter 'add', 'remove', 'view', or 'exit': exit