Python Project Assignment: Build Your Own Stock Watchlist App

Elio Rocha

Monday, May 12th, 2025

Objective

Create a command-line Python application using the `yfinance` library that allows a user to build and monitor a personalized stock watchlist. This project will help you practice working with user input, lists, dictionaries, and third-party APIs.

Project Description

Your program should allow the user to:

- Enter stock ticker symbols (e.g., `AAPL`, `TSLA`, `GOOGL`).
- Retrieve and display real-time data for those tickers using the `yfinance` library.
- View a simple watchlist summary showing the latest prices.

Core Requirements

1. Prompt the user to input one or more stock ticker symbols.
2. Use `yfinance` to fetch the current price for each symbol.
3. Display a clear table or list showing:
    - Stock Symbol
    - Current Price
4. Allow the user to add multiple stocks in one session.

Optional Features

If you want to push the project further, try adding the following features:

- Show additional data: company name, daily price change, or market cap.
- Save the watchlist to a file and load it the next time the program starts.
- Let the user remove a stock from their watchlist.
- Add an update loop to refresh prices every 30 or 60 seconds.

- Let the user view recent historical prices (e.g., last 5 days of closing prices).
- Create a menu-based interface (e.g., 1: Add stock, 2: View watchlist, 3: Quit).
- Calculate and display the total value of shares if the user owns some quantity of each stock.

Advanced Challenges

Take the project to the next level with these more complex features:

- **Plot a stock's historical performance**: Use libraries like `matplotlib` or `plotly` to show stock trends over time.
- **Email or SMS alerts**: Send alerts when a stock price goes above or below a user-defined threshold (e.g., using `smtplib` or `Twilio`).
- **Simulate a portfolio**: Let users buy and sell stocks using a virtual balance, and track their profit/loss over time.
- **Build a GUI version**: Use `tkinter` or `PyQt` to turn your CLI app into a desktop application.
- **Web scraping for news**: Display the latest news headlines for a stock using libraries like `BeautifulSoup` or `requests`.
- **Support multiple users**: Implement user login (no password required) and allow different users to maintain separate watchlists.

Learning Goals

- Practice using external libraries and APIs in Python.
- Work with user input, loops, and data structures like dictionaries and lists.
- Get comfortable thinking through project structure and expanding features over time.

Resources

- yfinance on PyPI
- Official yfinance GitHub repository
- Python documentation: docs.python.org/3