

Team Kappa:

Kanishk Kothakonda

Oliver Garnica

Kibrom Bereket

<https://github.com/olivergarnica/206-final-proj.git>

SI 206: Final Project Report

A. For this project, our goal was to collect and analyze financial and economic data using three different APIs. We planned to use:

- Finnhub API to get insider trading data for various companies
- Marketstack API to get historical stock prices for the same companies
- EconDB API to collect U.S. macroeconomic indicators like GDP, CPI, unemployment rate, and other indicators

Focusing on tech companies in the US, we wanted to gather insider trading activity, match it with the stock prices, and calculate the profit or loss from those trades. Also, we wanted to bring in economic indicators to help provide some context for the market around that time.

B. We used all three APIs as we planned:

- **Finnhub:** We pulled insider trades for high, mid, and low-cap companies like Apple, Microsoft, Google, and others. We collected trader names, transaction codes, prices, shares, and dates, and stored them in our database.
- **Marketstack:** We collected historical stock data for the same companies, including daily open and close prices, which we used to compute the 7-day price after each insider trade.

- **EconDB:** We gathered economic indicators like Real GDP, CPI, interest rates, and unemployment rates. These were saved in the macroeconomic_indicators table for analysis.

We calculated profit and loss for each insider trade and saved the results in a txt file. We also created visualizations like bar charts and scatter plots, which helped us better understand the patterns in insider trading.

C. Some problems we faced:

- The API key for Marketstack kept running out of API calls.
- Data from EconDB was from the 1940s instead of modern data. We had to make some changes to the way we were gathering data with the EconDB API.
- In the evaluation, we accidentally kept running more than the limit of 25 per call, which was pointed out in the evaluation, but we changed it since then.

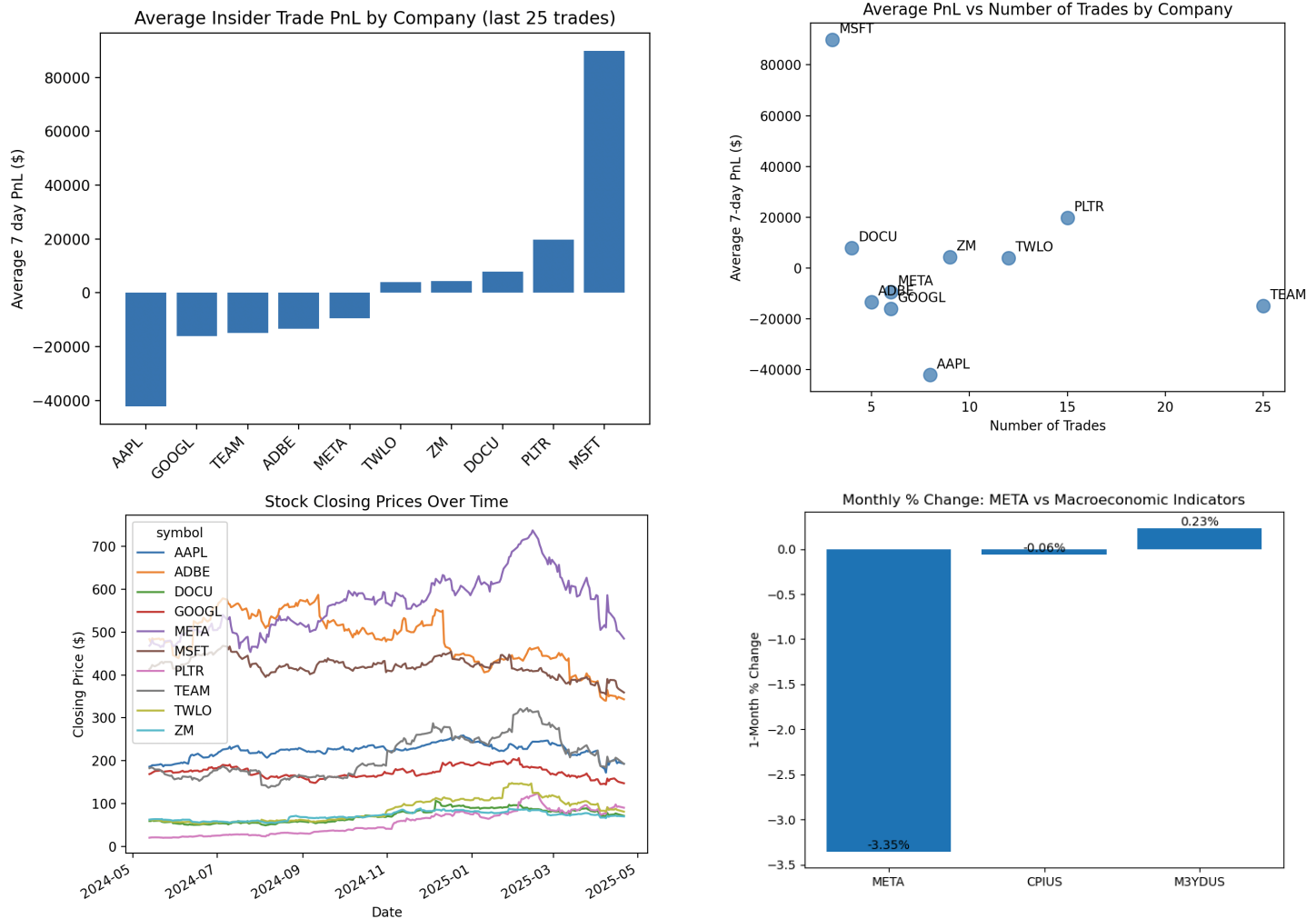
D. The images below show the results of the calculations we made for the profit and loss of each company and individual insider trade.

symbol	num_trades	avg_pnl
AAPL	8	-42184.53
ADBE	5	-13450.5
DOCU	4	7933.63
GOOGL	6	-16142.29
META	6	-9484.03
MSFT	3	89951.51
PLTR	15	19832.71
TEAM	25	-14996.66
TWLO	12	3955.52
ZM	9	4329.33

trade_id	insider_name	symbol	txn_code	transaction_date	txn_price	shares	price+7day	PnL
1	KONDO CHRIS	AAPL	S	2024-05-15	190.395	-4999	190.9	-2524.49
2	LEVINSON ARTHUR D	AAPL	S	2024-05-30	191.58	-75000	194.48	-217500.0
3	Adams Katherine L.	AAPL	S	2024-08-05	207.05	-2400	217.53	-25152.0
6	KONDO CHRIS	AAPL	S	2024-08-09	216.5	-5178	226.05	-49449.9
7	KONDO CHRIS	AAPL	S	2024-08-15	225.0	-8706	224.53	4091.82
35	WILLIAMS JEFFREY E	AAPL	S	2024-10-02	227.22	-8056	229.54	-18689.92
37	Adams Katherine L.	AAPL	S	2024-10-02	227.24	-2995	229.54	-6888.5
42	COOK TIMOTHY D	AAPL	S	2024-10-02	226.57	-7193	229.54	-21363.21
43	Althoff Judson	MSFT	S	2024-05-23	425.678	-25000	414.67	275200.0
46	Numoto Takeshi	MSFT	S	2024-05-31	416.6	-322	423.85	-2334.5
49	Numoto Takeshi	MSFT	S	2024-06-03	415.53	-244	427.87	-3010.96
71	HENNESSY JOHN L	GOOGL	S	2024-05-13	169.4974	-473	176.92	-3510.89
77	Pichai Sundar	GOOGL	S	2024-05-15	173.5725	-12496	176.38	-35082.52
86	O'Toole Amie Thuener	GOOGL	S	2024-06-03	173.86	-682	175.01	-784.3
87	Pichai Sundar	GOOGL	S	2024-06-05	177.5242	-3076	177.79	-817.6
90	HENNESSY JOHN L	GOOGL	S	2024-06-12	179.19	-11	176.3	31.79
133	Pichai Sundar	GOOGL	S	2024-06-20	177.7533	-7404	185.41	-56690.21
137	Zuckerberg Mark	META	S	2024-08-12	517.1535	-372	529.28	-4511.06
151	Zuckerberg Mark	META	S	2024-08-13	530.4561	-410	526.73	1527.7
172	Zuckerberg Mark	META	S	2024-08-14	532.91	-84	535.16	-189.0
213	Bosworth Andrew	META	S	2024-08-15	526.76	-5057	531.93	-26144.69
229	LI SUSAN J	META	S	2024-08-15	526.76	-4203	531.93	-21729.51
248	Clegg Nicholas	META	S	2024-08-15	526.76	-1133	531.93	-5857.61
289	Briggs Teresa	DOCU	S	2024-05-31	54.92	-499	52.05	1432.13

E. The first chart below shows the average PnL for each trade made by each company. The second chart shows the average PnL compared to the number of trades made by each company. The third chart shows the closing prices over the stocks that we chose over

the last year. The fourth chart (ADDED AFTER EVALUATION) shows the percentage change of our stock of choice vs two economic indicators (consumer price index and 3 month yield).



F. First, you need to create the database. To do so, run python main.py. This will fetch and insert all the data needed from our three APIs. To do our calculations, run python analysis.py. This will create two txt files called trade_pnl.txt and company_pnl_summary.txt, which will show all the results of our calculations. Finally, run python plots.py to get the charts we made.

G. analysis.py**a. calculate_and_write_pnl(db_path, output_path)**

- i. Input:
 1. db_path: path to SQLite database file (default "all_data.db")
 2. output_path: path to the .txt file to save PnL results
- ii. Output:
 1. Writes insider trade PnL data to a tab-separated text file
 2. No return value

b. analyze_pnls_by_company(pnl_file_path, output_path)

- i. Input:
 1. pnl_file_path: path to .txt file with trade PnL data
 2. output_path: path to .txt file to save summary statistics
- ii. Output:
 1. Writes number of trades and average PnL per company to a new file
 2. No return value

databases.py**a. __init__(self, db_path)**

- i. Input: db_path: path to the SQLite database
- ii. Output: Initializes database connection and creates tables

b. _create_tables(self)

- i. Input: None
- ii. Output: Creates all database tables if they don't already exist

c. insert_finnhub_data(self, transactions, symbol)

- i. Input:
 1. transactions: list of insider trade dictionaries
 2. symbol: stock symbol (string)
- ii. Output: Inserts up to 25 unique trades into the database
- iii. Returns: Nothing

d. insert_econdb_data(self, data, limit=25)

- i. Input:
 1. data: parsed JSON from EconDB API
 2. limit: max number of rows per indicator (default 25)
- ii. Output: Inserts trimmed economic data into the database

e. insert_marketstack_data(self, data, limit=25)

- i. Input:
 1. data: parsed JSON from Marketstack API
 2. limit: max number of entries (default 25)
- ii. Output: Inserts stock data into database

f. close(self)

- i. Input: None
- ii. Output: Commits changes and closes DB connection

econ.py**a. get_all_insider_trades(symbol)**

- i. Input: symbol: a stock ticker symbol (string)
- ii. Output: Returns a list of insider trade dictionaries for that symbol

marketstack.py**a. fetch_marketstack_data(symbol, limit, date_from, date_to)**

- i. Input:

1. symbol: stock ticker symbol
 2. limit: number of entries to fetch
 3. date_from/date_to: date range in YYYY-MM-DD format
- b. Output: Returns parsed JSON data with historical prices

insiders.py

- a. Contains the key and the 10 stock tickers of which we want to gather insider trading info for the past year
- b. Get_all_insider_trades:
 - I. send a request to api
 - II. extracts data
 - III. returns a list

main.py

- a. The drive that utilizes the different api python files to write information into the database.py class.

plot.py

- a. It has 4 uses (We added a fourth plot), and each one shows a different input of data.
- b. The fourth plot (new plot) requests user input to select a stock ticker and compare its price change in recent months vs two macroeconomic indicators: 3-month yield percentage change, and consumer price index.
- c. **Run this to get the plots**

H. Resources:

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
4/13/25	Not familiar with the insider trading transaction codes	Finnhub website	Yes. We finally learned purchases (P) and sales (S) are the transactions that really generate profit or loss. Hence, we did the PnL calculations accordingly.
4/13-4/22	Errors when pushing and pulling to and from git, and debugging	ChatGPT	Yes, it helped me to identify and fix errors.
4/13-4/22	Not familiar with all of the indicators and their labels	econdb.com	Yes, it did. I was able to find a bunch of different indicators with elaborate information on what each of them did.

4/02-4/22	Not completely sure on my sql documentation	sqlite.org and class slides	It did, was able to get a bunch of help understanding how to create separate tables and execute them
4/2	Completely confused on how to get started on the project	chatgpt	This helped me create ideas of how to organize the data into certain files and get the ball rolling
4/22	To better understand how to build a solid bar chart	matplotlib	Yes, helped me get the right functions to call and how to call them.
4/22	We had company symbols as duplicate strings in our database. Like APPL, APPL, MSFT, MSFT, ...	Professor Colleen	Yes, we created a new table named companies and gave each company an id.