

ITCS 6112

Software System Design and Implementation

Term Project

Group #1: Ara Chung, Abdullah Azihan, Chia Omotosho, Gavin Sidhu, Desmond

Williams

Project title: Coattails

Project Report Components

I. Abstract or Problem Description

Produce software that can predict market trends based on portfolios. We would like to create software that can collect public portfolios, like senators' portfolios, and recommend stocks that are predicted to do well. The prediction of trends will be for a time period that is to be determined.

II. Introduction: Explain the abstract in detail including what has been achieved.

Coattails is a financial investment tool for the interested everyday person, active intermediate trader and the stocks expert. *Coattails* is a web application with various features for analysis and visualization of investment data to better inform the users decision making. The focus of the analysis is on the financial portfolios of US Congressmen, with the input data being the stock investment details of the House of Representatives and the Senate members. The financial disclosures of US politicians are sought after because of their generally positive investment performances. According to

the *NY Times* 'Not-So-Representative Investors' essay "House members 'earn statistically significant positive abnormal returns,' outperforming the market by 6 percentage points. Senators do even better... Senate portfolios 'show some of the highest excess returns ever recorded over a long period of time, significantly outperforming even hedge fund managers,' with gains that are 'both economically large and statistically significant'"(Schwartz, 2011). With *Coattails* public data is manipulated and ideally to be presented to the user through the web application interface. Users can create a profile and log in to view the text based recommendations, time series of the specific stock assets and graph of good buys specific to the bodies of congress. The back end components of these functionalities are created but further integration on the front end is needed. Ideally additional features of the web application would be for users to save their favorite portfolios and financial asset links. Users should be able to link directly to the trading platform that they currently use or view options and select one to embed in their profile as a quick access point. There would also be a search and or index image page of the tickers and their corresponding asset/firm descriptions. This is so that upon gaining the information to make wiser decisions they can then do so easily. *Coattails* is intended to be a dynamic tool for investment learning.

III. Sample runs with proper output and or diagrams

```

i = 0
for companies in buy_list:
    for element in companies['Date']:
        for j in range(5202):
            try:
                if (trans_2021['transaction_date'][j].month in range(y(element)-1,y(element)+1)) & (trans_2021['ticker']
                    print('In %d, %s was a good buy' %(trans_2021['transaction_date'][j].month,stonks[i]))
            except:
                pass
        i = i + 1

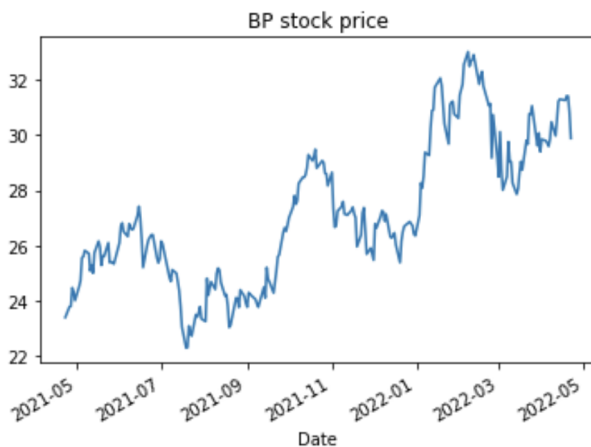
```

```

In 1, BP was a good buy
In 9, BP was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy

```

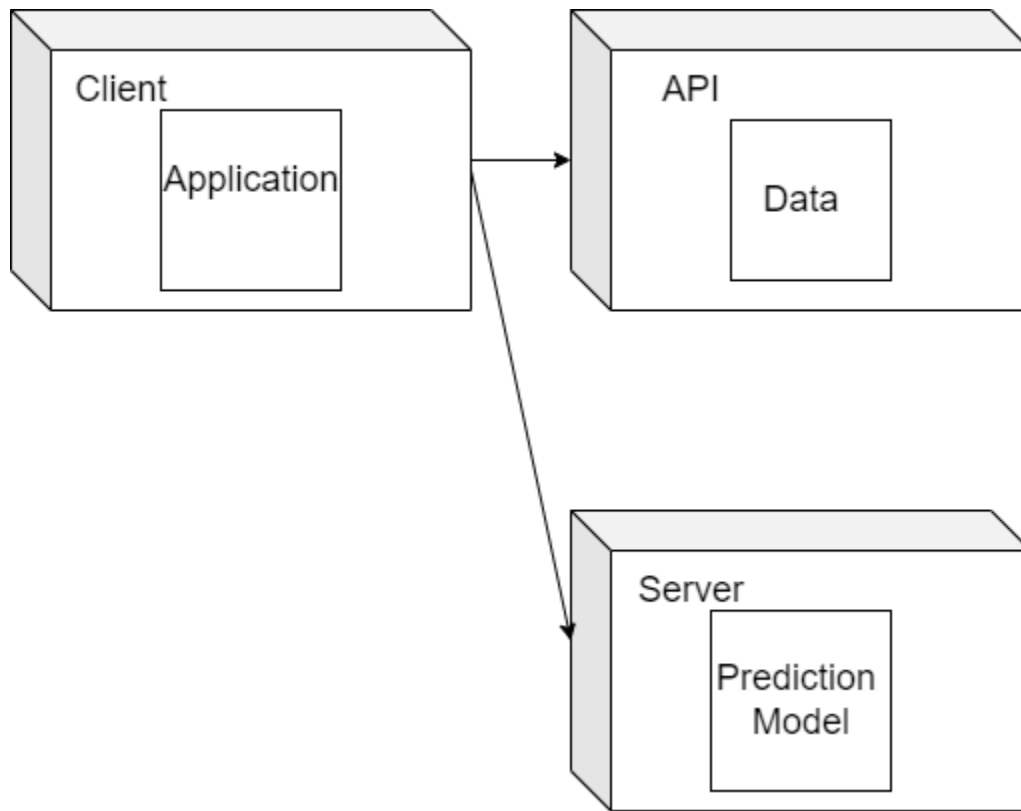
<AxesSubplot:title={'center':'BP stock price'}, xlabel='Date'>



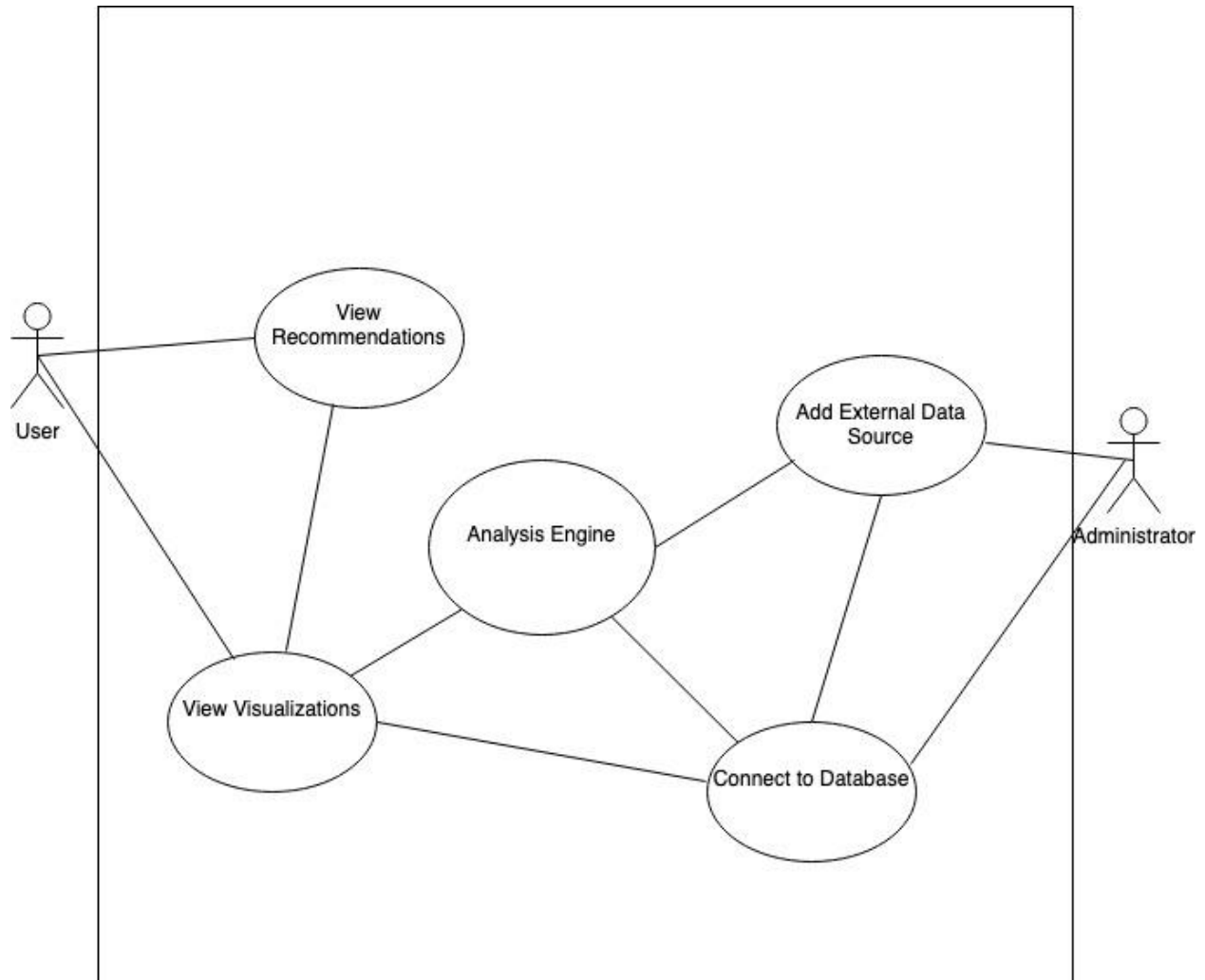
One of the key functionalities of our project is to be able to deliver insights on stock in 2 primary ways. They are a text based recommendation system and also a more traditional data visualization system(graphs). The results for these from a sample run are shown above.

Diagrams:

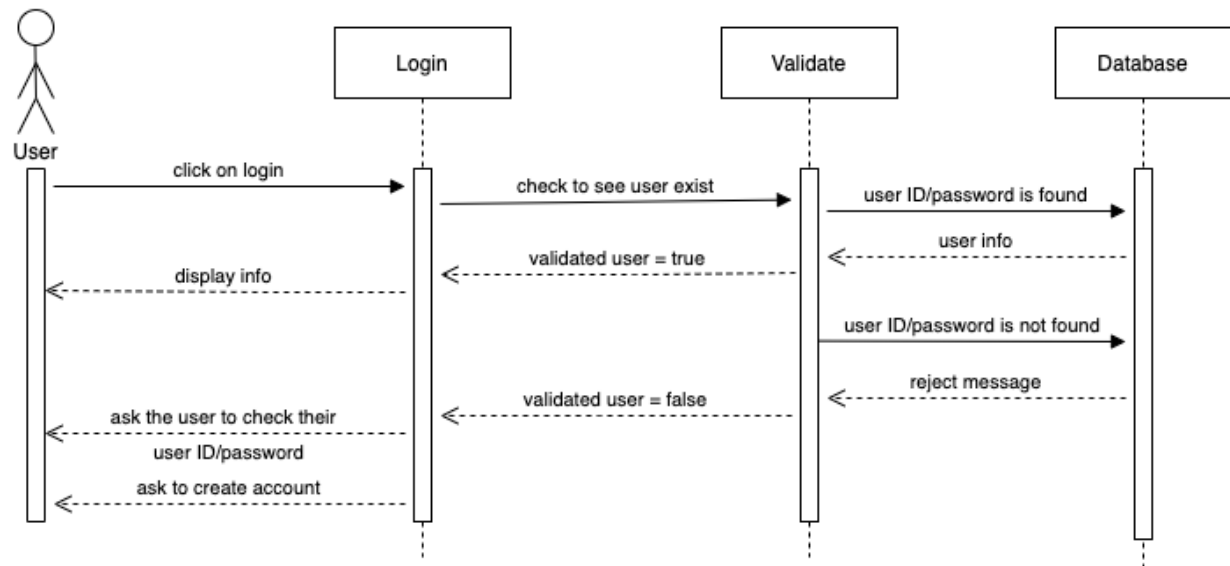
a. Deployment Diagram



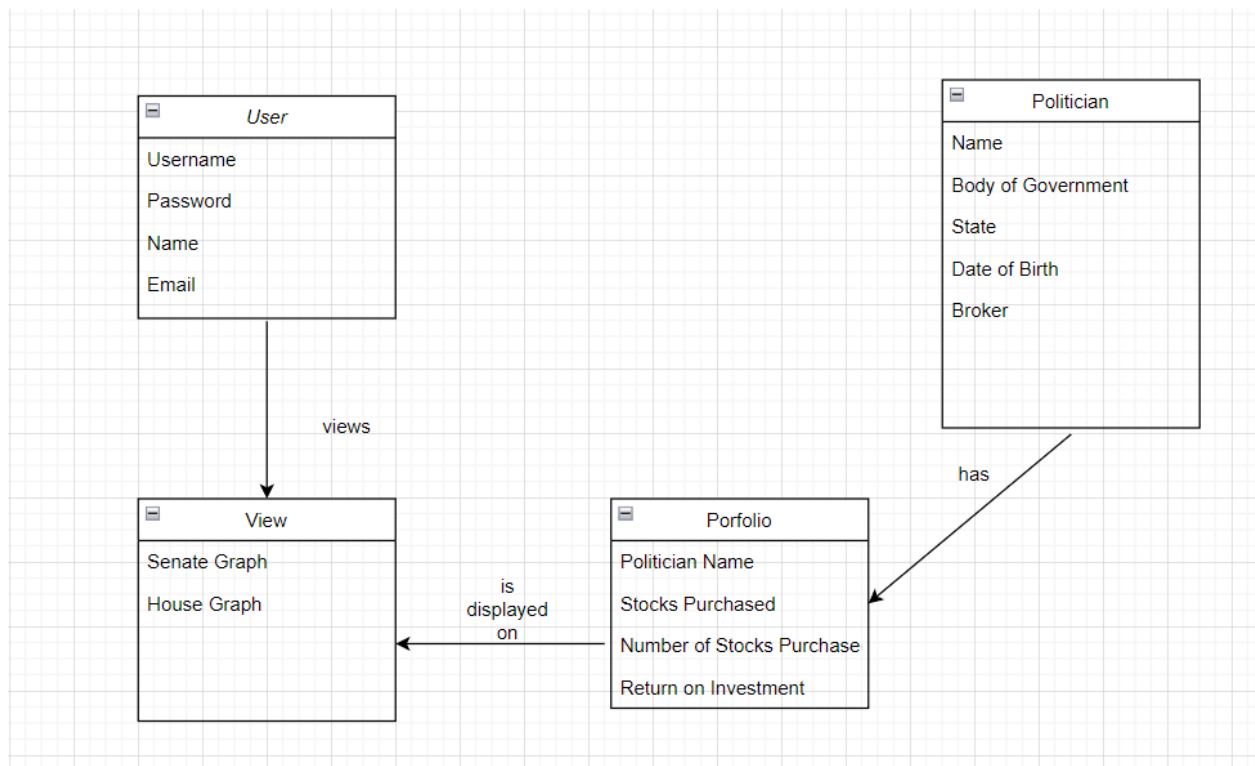
b. Use case Diagram(s)



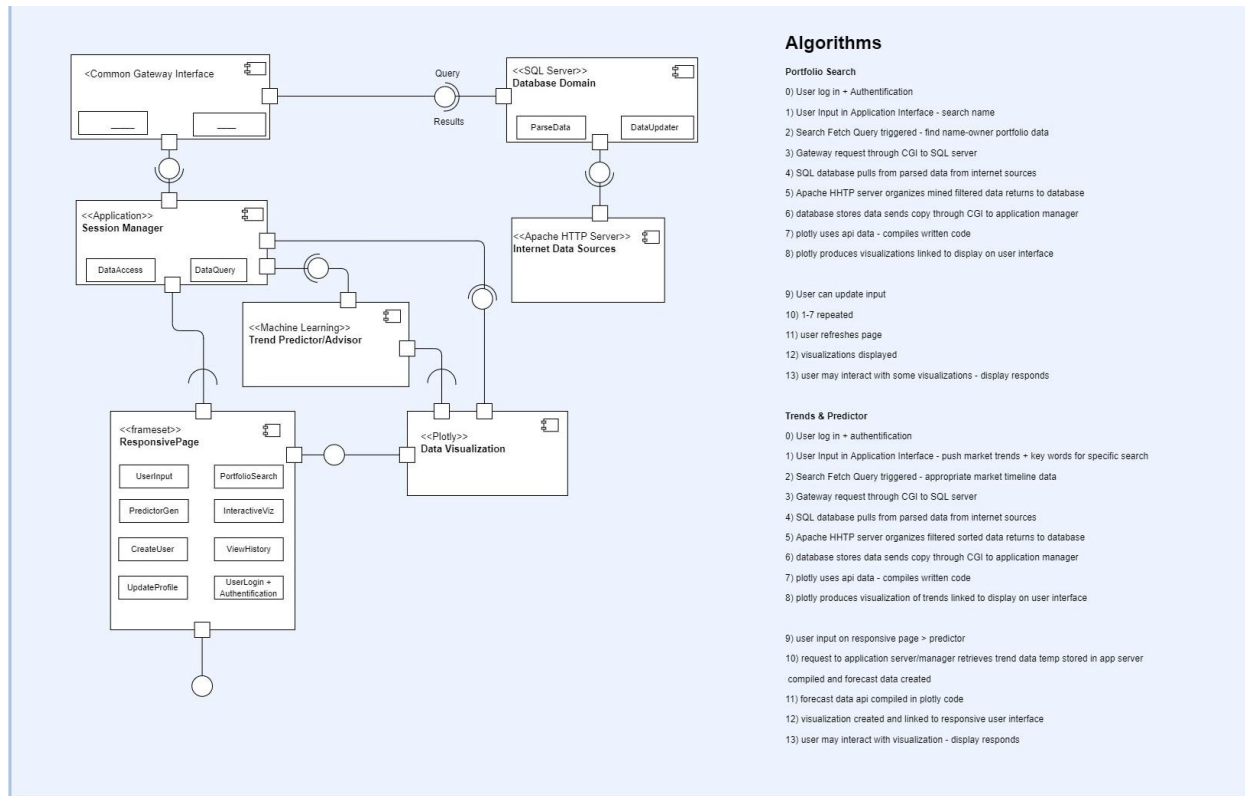
c. Sequence diagram



d. Complete class diagram with all the classes and their relationships.



e. Component diagram including the algorithm(s) behind your implementation if applicable.



Technical documentation:

a. Programming language:

- Python
- CSS & HTML: in the future to create a front end web application from scratch that may better integrate all the components we have

b. Libraries

- Plot.ly
- Pandas
- Numpy

- Streamlit
- Datetime
- Yahoo Financials: This is where our data is from.
- Congressmen Data : <https://senate-stock-watcher-data.s3-us-west-2.amazonaws.com>, also imported in code as downloaded csv file.

c. Tools and environments:

- Github: coding management
 - <https://github.com/achung3/coattails>
- googleDoc: storing documents
- Colab: storing our code
 - <https://colab.research.google.com/drive/1my0-OCHjez3xyoZE0opMS6RnL6kpqYNe?authuser=1>
- Bubble: used to create our website
 - coattails.bubbleapps.io/version-test/
- Draw.io: to create diagrams

Product backlog


- I. Refine your user stories taking into account the instructor's feedback. Break down previously identified large user stories (epics). Indicate which epics resulted in what new user stories. Estimate the size of your user stories. Use Fibonacci numbers within the range of 1 to 8 to represent a relative size of each user story. Label each user story as high, medium, or low priority. Note the cumulative size of all user stories in your product backlog.
 - High: 5, Finish the documentation for Term Project

- Low: 1, Create github and upload all our files/codes
- High: 2, Code: allow our code to push recommendations. Recommendations should be visible.
- Medium, 2, Website: Clean up the website, this will be used to allow users know what the end result will look like.

II. Describe the functionality that your partially implemented system will have at the end of this sprint

Have our code analyze the data and push recommendations. The recommendations should be visible for users to see and use that to make wise investment decisions.

III. Design key features of the user interface; provide sketches of your designs


[Contact Us](#)
[FAQs](#)
[SIGN UP OR LOGIN](#)


Analyze

Well known investment portfolio to make your next move

This page has everything you need to get started. If you're already a Bubble Pro, you can use parts of this boilerplate, or delete the whole page.

[GET STARTED](#)

Current V1.0.0



A collection of building blocks you can use to get started. Customize and experiment!



Create Account

Begin your journey by creating your account.

Start Your Journey

Create Account

Continue Your Journey

Login



Analyze Data

Form fields and inputs are essential parts of collecting data from your users. Here are a few examples to get you started.

Analyze Portfolio

Search Portfolio



Grow Your Investment

Bring your site to life with visual in-page elements and images. You can use static uploaded visuals, or dynamic data driven ones.



Coattails Footer

Read and Learn

[All About Coattails](#)

[Reference Library](#)

[Template Library](#)

Connect and Ask

[Coattails Forum](#)

[Trusted Agencies](#)

[FAQ's](#)

Bookmark and Follow

[Coattails status](#)

[Twitter](#)

[Blog](#)

Copyright Coattails 2022

[About the Company](#)

[Terms and conditions](#)

[Reset password](#)

Sprint reports

I. What functionality does the system have at the end of this sprint?

Have our code analyze the data and push recommendations. The recommendations should be visible for users to see and use that to make wise investment decisions.

II. List the user stories that you successfully implemented during this sprint.

- Finish the documentation for Term Project
- Create github and upload all our files/codes
- Code: allow our code to push recommendations. Recommendations should be visible. Allow our code to create visualizations of the data for user interpretation.
- Website: Clean up the website, this will be used to allow users know what the end result will look like.

III. Did you end up making any changes to any of these user stories? Did you break down any further user stories?

No, because it was our last sprint therefore there wasn't much to change.

IV. Did you identify any new user stories during this sprint and, if so, did you add them to the product backlog or decide to implement them right away?

We didn't think about adding all our information in github, when we thought of new tasks we implemented right away if it was possible within this sprint. However, the tasks we didn't have time for were added to the backlog for future sprints.

V. What are the "lessons learned" at the end of this sprint?

Finding the right data quality is very important, not all data is equal. Depending on the data our recommendations can shift therefore finding the high quality data is valuable.

VI. What would you do differently next time?

- Start earlier, it's always better to have more time than less.
- Assign tasks and make sure each task is completed within the sprint.
- Create a more cohesive outline of the front end and back end pieces to put together, rather than working on the pieces then figuring out how to combine everything.

- Having solid goals, we were confused on what we wanted so creating the next task was difficult.

VII. Are there any user stories left unimplemented in the backlog?

We finished all the tasks for this sprint, however if we had more time we would like to improve our product. We believe that great products need consistent modifications, therefore it will have endless backlogs.

User manual

- I. Detail all necessary steps needed to deploy/install your system. Provide all necessary technical specifications.

Deployment is handily contained in a linear Python notebook fashion with the cells run one by one in order corresponding to the following tasks:

- 1) Import dependant libraries and datasets

- 2) Load, organize and format data from both yahoo finance library and

congressperson transaction dataset in data frames

- 3) Run nearest neighbors like comparison algorithm that prints text based results

- 4) Re-access yahoo finance database to present graphical information on stocks of

interest

- II. Explain the main features of the system to a potential user who may not be familiar with it.

Create an account:

follow the prompts to fill in your credentials and save

Log in account:

fill in your credentials to view view your Coattails account profile

Customize account profile :

Add a quick link to your favorite profiles and view them, organize them, and another link to your trading platform.

Search in the search bar:

type in political figures name to search for their portfolio, you may also filter by ticker.

View “good buy” recommendations:

via the indicator graphic, this is generated in two sets. 1 from general Yahoo finance data and another from the congress’s mined 2021’s financial data

View:

Future: Index reference page of ticker symbols and the asset or firm name

View the data:

visualization of the financial data pulled

Link:

Direct link to the trading platform that you use so you can grow your investments.

- III. Provide a walkthrough for the main scenario of using your system; include screenshots as necessary

LOADING DATA(CONGRESS)

```

1 congress_trans = pd.read_csv("/content/drive/MyDrive/all_transactions.csv")
2 #https://housestockwatcher.com/api
3 #https://senatestockwatcher.com/api

```

```

1 congress_trans.columns

```

```

Index(['disclosure_year', 'disclosure_date', 'transaction_date', 'owner',
      'ticker', 'asset_description', 'type', 'amount', 'representative',
      'district', 'ptr_link', 'cap_gains_over_200_usd'],
      dtype='object')

```

```

1 congress_trans

```

| | disclosure_year | disclosure_date | transaction_date | owner | ticker | asset_description | type | amount | representative |
|---|-----------------|-----------------|------------------|-------|--------|-------------------------|----------|--------------------|--------------------|
| 0 | 2021 | 10/04/2021 | 2021-09-27 | joint | BP | BP plc | purchase | \$1,001 - \$15,000 | Hon. Virginia Foxx |
| 1 | 2021 | 10/04/2021 | 2021-09-13 | joint | XOM | Exxon Mobil Corporation | purchase | \$1,001 - \$15,000 | Hon. Virginia Foxx |

LOADING DATA(BUY RATINGS)

```

1 stonks = list((trans_2021['ticker'].unique()))

```

```

1 bunches = []
2 for items in stonks:
3     bunches.append(yf.Ticker(items))

```

```

1 i = 10
2 buy_ratings2 = []
3
4 for rating in bunches:
5     if i > 0:
6         buy_ratings2.append(rating.recommendations)
7     i = i - 1

```

```

1 buy_ratings2[0]

```

| | Date | Firm | To Grade | From Grade | Action |
|---|------------|-------------------|----------------|------------|--------|
| 0 | 2012-05-23 | Canaccord Genuity | Buy | | up |
| 1 | 2012-08-01 | Jefferies | Hold | Buy | down |
| 2 | 2012-12-14 | Credit Suisse | Neutral | Outperform | down |
| 3 | 2013-01-08 | Raymond James | Market Perform | Outperform | down |

RUN RECOMMENDATION ALGORITHM

```

1 i = 0
2 for companies in buy_list:
3     for element in companies['Date']:
4         for j in range(5202):
5             try:
6                 if (trans_2021['transaction_date'][j].month in range(y(element)-1,y(element)+1)) & (trans_2021['ticker'][j] == st
7                     print('In %d, %s was a good buy' %(trans_2021['transaction_date'][j].month,stonks[i]))
8             except:
9                 pass
10        i = i + 1
11

```

```

In 1, BP was a good buy
In 9, BP was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy
In 6, BLK was a good buy

```

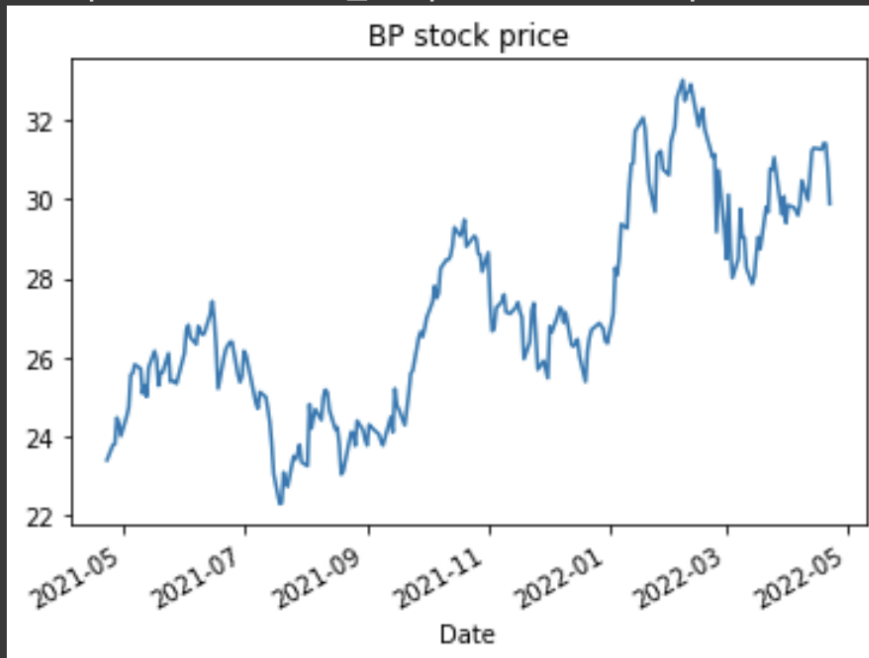
CHECK VISUALIZATIONS

```

1 ticker = yf.Ticker("BP");
2 bp_df = ticker.history(period = "1y")
3 bp_df['Close'].plot(title = "BP stock price")

```

<matplotlib.axes._subplots.AxesSubplot at 0x7f09a46e8ad0>



WALKTHROUGH

1)Assure 3 different datasets are present in same directory as code: “all_transactions.csv”, “all_transactions_senate.csv” and “all_transactions_house.csv”

2)Open CoattailsBack.ipynb in either jupyter-notebook or google colab

3) Install necessary packages/libraries

-this should be achieved simply by running the 2nd cell in Coattails.ipynb

-but the necessary packages are:

Numpy, Pandas, yahoo finance, yahoo financials

4)The ease of the notebook format means the cells just need to be run one by one to produce the desired sample output

IV. Provide walkthroughs for at least two additional scenarios with additional/alternative functionality; include screenshots as necessary

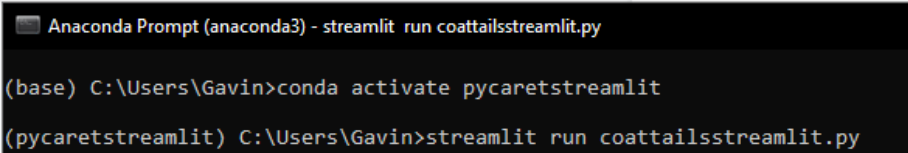
PT.1

1) Same as the above walkthrough, ensure that the datasets are present in the same directory

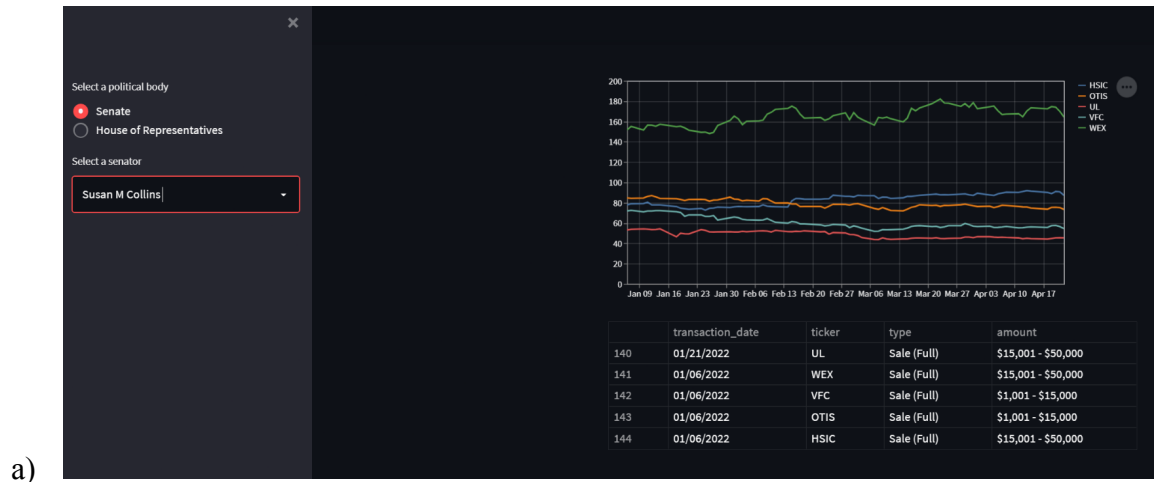
2) Make sure the necessary libraries and packages are loaded and have the correct versions (you need python 3.8 or older, streamlit, pandas, and datetime in addition to what is above)

3) Make sure you are in that directory and that CoattailsStreamlit.py is as well

4) Open a command prompt and type in “streamlit run CoattailsStreamlit.py”

a) The screenshot shows a terminal window titled "Anaconda Prompt (anaconda3) - streamlit run coattailsstreamlit.py". It displays the following commands and their outputs: 1. The prompt is "(base) C:\Users\Gavin>". 2. The command "conda activate pycaretstreamlit" is entered, and the prompt changes to "(pycaretstreamlit) C:\Users\Gavin>". 3. The command "streamlit run coattailsstreamlit.py" is entered, and the prompt returns to "(base) C:\Users\Gavin>".

- b) In the above screenshot, I have used a virtual environment to use the right versions of the packages needed for everything to work together and you will notice that I am in the directory that I have the files on for on my computer
- 5) The visualization should load in a localhost and show the stock portfolios of the politician that you have selected from a particular political body



PT 2

- 1) For the “House Investments Visualization and Buy Statements” the program is written in python using a notebook compiler. Ideally it will be translated into a formal .py script which can be run like the above. *Note to delete: This may or may not be added in a few hours i’ve been working on it*
- 2) As above Make sure the stated data files (provided in github) are made available in the same directory. “Profitable_house_assets.csv”, and “house_Good_Investments_Sorted”
- 3) Open ‘Coattails_House_Investment_Notebook_Visualizations.ipynb’ in either jupyter-notebook or google colab or a notebook of your choice.
- 4) Install and import necessary packages/libraries

-this should be achieved simply by running the 2nd cell in Coattails.ipynb

- necessary packages for this particular visualization are as follows:

Plotly - plotly.express as px, plotly.graph_objects as go, chart_studio.plotly, chart-studio.plotly, IPython.display import html as html_print. *The later 5 are specifically for the exportation of the visualizations into a file or web page which has so far not been achievable because a wsgi server is needed. This is the same with the dash installed.*

Pandas - pandas as pd, pandas_datareader.data as web,

Numpy - numpy as np , *used in earlier versions of script. Not necessary now.*

HOUSE INVESTMENTS VISUALIZATIONS

Visualizations script notebook form

Notes:
un-comment the installations if needed
replace csv export paths

Visualize in notebook output *not wrapped in function

```
In [1]: #!pip install plotly
        #pip3 install numpy
        #!pip install chart_studio
        #!pip install pandas_datareader
        #!pip install jupyter-dash
        #!pip install yfinance
        #!pip install yahoofinancials
        #!pip install dash
        #!pip install chart-studio
        import plotly.express as px
        import chart_studio.plotly as py
        import plotly.graph_objects as go
        import pandas_datareader.data as web
        import pandas as pd
        import numpy as np
        from IPython.display import HTML as html_print
```

5) notebook format means the cells just need to be run one by one to produce the desired sample output as follows

6) In cell 2 read in necessary data files (csv's) as dfhg and dfhgcc

```
In [2]: #import/read these files
        #To be imported as dfhg and value counted into my_dict -> for House Investment Statements
        dfhg = pd.read_csv("profitable_house_assets.csv")
        #TO BE imported and USED FOR px.bar figure
        dfhgcc = pd.read_csv("house_Good_Investments_Sorted.csv")
        #call the data frame not printed so it retains the correct organization of rows columns
```

- 7) Run cell 3 which takes the data that represents all the significantly gainful financial investments of 2021 for the House of Representatives and numbers the frequency of a particular asset/firm ticker in that list.

```
In [3]: #counts frequency of asset within the list
dfhg.value_counts(subset=None, sort=True, ascending=False, dropna=True)

Out[3]: ticker  asset_description
AMZN  Amazon.com Inc          4
V      Visa Inc              4
IFNNY  Infineon Technologies AG  4
HDS     HD Supply Holdings      4
BLL     Ball Corporation        3
..
FANUY  Fanuc Corp ADR          1
FANG    Diamondback Energy Inc  1
F        Ford Motor Company     1
EXPE    Expedia Group Inc       1
ZUO     Zuora, Inc. Class A     1
Length: 371, dtype: int64
```

- 8) Some asset descriptions are written differently so the ticker is used to append or flatten.
- Run cell 4 to create the dictionary of keys and values with the counter data

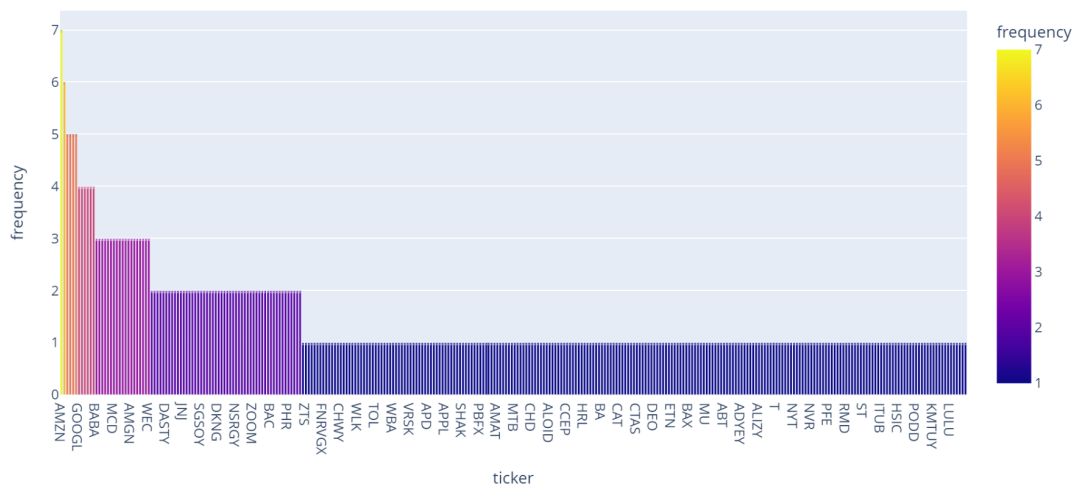
```
In [4]: #turns the counter into a dictionary with named ticker value and the frequency number in the pair
#This dict will be used for the FREQUENCY GRAPH
from collections import Counter
my_dict = dict(Counter(dfhg.ticker))
my_dict
```

```
Out[4]: {'COLD': 2,
'SEP': 1,
'DXCM': 2,
'HD': 3,
'ICUI': 2,
'IFNNY': 6,
'PHR': 2,
'SITE': 2,
'VCYT': 3,
'AMZN': 7,
'DOMO': 1,
'DIS': 2,
'GOOGL': 4,
'GOOG': 5,
'AMGN': 3,
'AAPL': 5,
'CFG': 2,
'INTC': 3,
'INTL': 1,
```

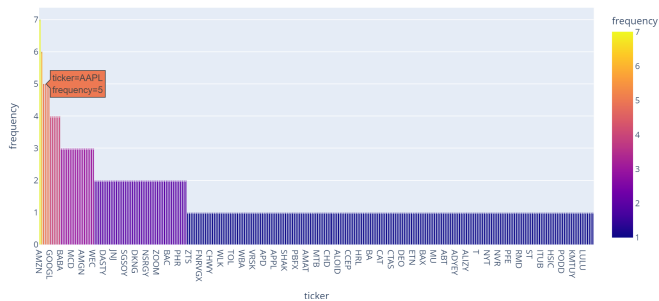
9) Run cell 5 to see the final chart visualization of the “House of Reps 2021 Good Investments With Frequent Gain”.

```
In [5]: #import House_Good_Investments_Sorted.csv as dfhgcc
#INSEERT PD DATA READ HERE
houseGoodInvestmentsFig = px.bar(dfhgcc, x='ticker', y='frequency', title='House of Reps 2021 Good Investments With Frequent Gain',
houseGoodInvestmentsFig
```

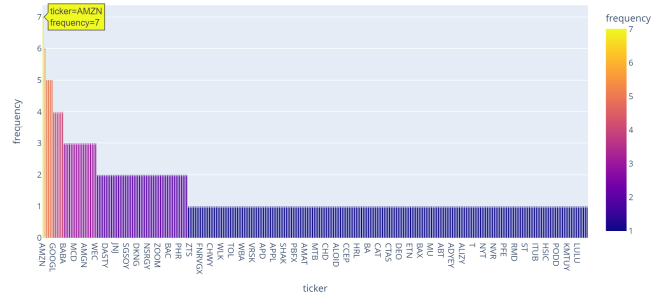
House of Reps 2021 Good Investments With Frequent Gain



House of Reps 2021 Good Investments With Frequent Gain



House of Reps 2021 Good Investments With Frequent Gain



note the chart is interactive - by hovering pointer. note all ticker symbols included in the data are not labeled on chart (x).

10) Run cell 6. Since all ticker symbols included in the data are not labeled on chart (x). It is necessary to provide a written reference. Ideally users could search for the ticker symbol of the financial asset /firm of their interest as displayed. *Note colors and frequency groupings correspond with the graph as they would be displayed on the same “grow your investment” webpage and linked to portfolios*

```
In [6]: #WRITEN GOOD BUYS SUGGESTIONS

keys7 = [k for k, v in my_dict.items() if v==7]
print('\n\033[2;33m The Best buys are/is:\n', keys7)

keys6 = [k for k, v in my_dict.items() if v==6]
keys5 = [k for k, v in my_dict.items() if v==5]
print('\n\033[38;2;255;165;0m Great buys are/is:\n', keys6, keys5)

keys4 = [k for k, v in my_dict.items() if v==4]
print('\n\033[2;31m Great buys are/is:\n', keys4)

keys3 = [k for k, v in my_dict.items() if v==3]
print('\n\033[2;35m Very good buys are/is:\n', keys3)

keys2 = [k for k, v in my_dict.items() if v==2]
print('\n\033[2;35;1m Really good buys are:\n', keys2)

keys1 = [k for k, v in my_dict.items() if v==1]
print('\n\033[2;34m Pretty good buys are/is:\n', keys1)

The Best buys are/is:
['AMZN']

Great buys are/is:
['IFNNY'] ['GOOG', 'AAPL', 'V', 'WMT']

Great buys are/is:
['GOOGL', 'MSFT', 'TSM', 'KTOS', 'SPCE', 'HDS']

Very good buys are/is:
['HD', 'VCYT', 'AMGN', 'INTC', 'PYPL', 'BLL', 'BRK.B', 'FB', 'MA', 'PG', 'TSLA', 'BXS', 'BABA', 'MRK', 'WEC', 'D', 'TLRY', 'MCD', 'W']

Really good buys are:
['COLD', 'DXCM', 'ICUI', 'PHR', 'SITE', 'DIS', 'CFG', 'QCOM', 'SBUX', 'ROP', 'INFO', 'FISV', 'ECL', 'CACI', 'ADBE', 'AAGIY', 'BAC', 'BIIB', 'CHTR', 'CMS', 'COST', 'DASTY', 'EW', 'FDX', 'IBM', 'JNJ', 'LZAGY', 'MSI', 'NSRGV', 'NEM', 'NOC', 'NVDA', 'OK E', 'RIO', 'RHBY', 'SPGI', 'SAP', 'SGSOY', 'STT', 'SSMX', 'TFC', 'USB', 'UNH', 'DKNG', 'MCO', 'RSG', 'AM', 'FLEX', 'VIAC', 'ALXN', 'ZOOM', 'BKEP']

Pretty good buys are/is:
['SEP', 'DOMO', 'INTL', 'LMT', 'MU', 'TMUS', 'GO', 'DELL', 'CHNG', 'CDW', 'ABT', 'ABBV', 'ACHC', 'ATVI', 'ADDDY', 'AMD', 'A DYEY', 'A', 'AL', 'ALC', 'ALFVY', 'ADS', 'ALIZY', 'ABEV', 'AMT', 'AWK', 'ANTM', 'AON', 'T', 'ATH', 'ATLKY', 'BAX', 'BBY', 'B A', 'BSX', 'AVGO', 'CDNS', 'CNI', 'CCL', 'CAT', 'SCHW', 'CHKP', 'CHGCY', 'CI', 'XEC', 'CTAS', 'CMCSA', 'CLGX', 'CCK', 'DHR', 'DBSDY', 'DEO', 'FANG', 'DFS', 'DG', 'DXC', 'EMN', 'ETN', 'EMR', 'EHC', 'EPKY', 'EQIX', 'EXPE', 'XOM', 'FANUY', 'FITB', 'BE N', 'FUPBY', 'GLPI', 'GPN', 'HDB', 'HSIC', 'HRC', 'HSBC', 'HUN', 'IBN', 'IDXXX', 'PODD', 'ISRG', 'ITUB', 'JBL', 'JPM', 'KKR', 'KMTUY', 'KUBTY', 'AIQUY', 'LYV', 'LRLCY', 'LOW', 'LULU', 'LYB', 'MMC', 'MTCH', 'MCHP', 'MS', 'MOS', 'MSCI', 'NTAP', 'NFLX', 'NYT', 'NEE', 'NKE', 'NCLTY', 'NVZMY', 'NUE', 'NVR', 'OKTA', 'OMF', 'ORCL', 'ORLY', 'PH', 'PEP', 'PFE', 'PSX', 'PNGAY', 'PN C', 'PLD', 'RMD', 'ROK', 'SC', 'SLB', 'SBGSY', 'SEE', 'ST', 'SNAP', 'SONVY', 'SIVB', 'SYIEY', 'SYF', 'SNPS', 'TGT', 'TCEHY', 'TMO', 'TOL', 'TRU', 'UBER', 'UNICY', 'UL', 'UNP', 'UPS', 'UNVR', 'VRSK', 'VTRS', 'VMC', 'WBA', 'WLK', 'WMB', 'WYND', 'WH', 'YNDX', 'ZG', 'Z', 'ZTS', 'FNRVGX', 'HII', 'NGLS', 'HAE', '--', 'ADDDY', 'APD', 'BPHSPX', 'CLX', 'HUM', 'LIN', 'RTX', 'VRTX', 'CLYCT', 'LDOS', 'SAIC', 'TMUBMUSD03M', 'ARW', 'CCEP', 'CMP', 'COP', 'F', 'HRL', 'LB', 'MTB', 'PRU', 'PSA', 'VFC', 'WM', 'APP L', 'STZ', 'GTN', 'MCS', 'QCHR', 'SHAK', 'HSIP', 'ENLC', 'NS', 'PBFX', 'USAC', 'USDP', 'GLD', 'COHR', 'KR', 'KMI', 'SPXC', 'B XS$A', 'AMAT', 'DNKN', 'ROKU', 'SJCOC', 'ZBRA', 'SIRI', 'PRAH', 'LH', 'CSGP', 'AME', 'ALOID', 'FCEL', 'PLUG', 'CHD', 'ADI', 'ASML', 'FSLR', 'MJNA', 'ICE', 'CHWY', 'DOCU', 'GM', 'ARNC', 'TDG', 'ZUO', 'BHP', 'EQNR', 'NCMGF', 'WOLWF']

The best buys are/is:
['AMZN']

Great buys are/is:
['IFNNY'] ['GOOG', 'AAPL', 'V', 'WMT']

Great buys are/is:
['GOOGL', 'MSFT', 'TSM', 'KTOS', 'SPCE', 'HDS']

Very good buys are/is:
['HD', 'VCYT', 'AMGN', 'INTC', 'PYPL', 'BLL', 'BRK.B', 'FB', 'MA', 'PG', 'TSLA', 'BXS', 'BABA', 'MRK', 'WEC', 'D', 'TLRY', 'MCD', 'W']

Really good buys are:
['COLD', 'DXCM', 'ICUI', 'PHR', 'SITE', 'DIS', 'CFG', 'QCOM', 'SBUX', 'ROP', 'INFO', 'FISV', 'ECL', 'CACI', 'ADBE', 'AAGIY', 'BAC', 'BIIB', 'CHTR', 'CMS', 'COST', 'DASTY', 'EW', 'FDX', 'IBM', 'JNJ', 'LZAGY', 'MSI', 'NSRGV', 'NEM', 'NOC', 'NVDA', 'OK E', 'RIO', 'RHBY', 'SPGI', 'SAP', 'SGSOY', 'STT', 'SSMX', 'TFC', 'USB', 'UNH', 'DKNG', 'MCO', 'RSG', 'AM', 'FLEX', 'VIAC', 'ALXN', 'ZOOM', 'BKEP']

Pretty good buys are/is:
```

*A query call / search bar ideally
would be used within the web page
or simply ‘ctrl f’*

IV. Conclusions

This project was an eye opener for us on many fronts as we learned the many intricacies of stock trading, and how much information is actually available on public figures. On a coding aspect we learned more about how to read data from an API and system interactions. More importantly we were taught important strategies for software development. As a group we successfully utilized Sprint Techniques and properly recorded our progress. Below is our documentation and user manual for our product.