

# FINANCIAL ANALYSIS PRESENTATION

+  
BY

ENITA OMUVWIE

# Company Profile

- + This is the first page of my application
- + The first thing you need to do is to select a ticker
- + Then it gives you everything about the company with the ticker assigned

**YAHOO FINANCIAL DASHBOARD**

Data source: Yahoo Finance

**Tab 1 - COMPANY PROFILE**

	Value
zip	76155
sector	Industrials
fullTimeEmployees	119800
compensationRisk	4
auditRisk	8
longBusinessSummary	American Airlines Group Inc., through its subsidiaries, operate...
city	Fort Worth
phone	817 963 1234
state	TX
shareHolderRightsRisk	7
compensationAsOfEpochDate	1609372800
governanceEpochDate	1632614400
boardRisk	6
country	United States
website	http://www.aa.com
maxAge	86400
overallRisk	6
address1	1 Skyview Drive
industry	Airlines

# Summary

- + This is the second page of the application
- + This contains all attributes of current day trading from Volume, Day's range, Open price, Market Cap e.t.c
- + You can download and create file name to directly download to your system
- + It has a feature of update as well to get information updated

**YAHOO FINANCIAL DASHBOARD**

Data source: Yahoo Finance

**Tab 1 - Summary**

	0
1y Target Est	19.08
52 Week Range	12.23 - 26.09
Ask	20.31 x 4000
Avg. Volume	29535651.0
Beta (5Y Monthly)	1.68
Bid	20.30 x 1300
Day's Range	20.06 - 21.35
EPS (TTM)	-5.18
Earnings Date	Jan 26, 2022 - Jan 31, 2022
Ex-Dividend Date	Feb 04, 2020
Forward Dividend & Yield	N/A (N/A)
Market Cap	13.138B
Open	21.34
PE Ratio (TTM)	nan
Previous Close	21.33
Quote Price	20.290000915527344
Volume	48375401.0

# Summary

- + This is also on the second page
- + It gives you a line graph containing Low, High, Volume e.t.c of a selected stock for a selected period or interval



# Chart

- + This is the third page of the application containing a graph showing the high, low, open, close, volume of the selected stock in a specified period or time frame
- + This graph consists of three different plots: Candlestick, Line and Bar graph
- + The bar holds the volume, the ohlc holds the high, low, closing and opening prices of the stock
- + The Line graph shows the moving average which helps in smoothing out the variations to indicate unknown patterns in the data
- + Reference of code: <https://asxportfolio.com/shares-python-for-finance-plotting-stock-data>



# Statistics

- + This tab shows us the statistics of each stock within a given period
- + This shows the Valuation Measures, Financial Highlights and Trading Information of a selected stock

×

Pick your assets

AAL

Period

1M

Select tab

☐ Company profile

☐ Summary

☐ Chart

☒ Statistics

☐ Financials

☐ Analysis

☐ Monte Carlo Simulation

☐ Total Traded Analysis

Statistics

Valuation Measures

	0	1
0	Market Cap (intraday) 5	13.81B
1	Enterprise Value 3	45.26B
2	Trailing P/E	2.81
3	Forward P/E 1	1.11k
4	PEG Ratio (5 yr expected) 1	<NA>
5	Price/Sales (ttm)	0.52
6	Price/Book (mrq)	<NA>
7	Enterprise Value/Revenue 3	1.85
8	Enterprise Value/EBITDA 7	-18.69

Financial Highlight

Fiscal Year

	Attribute	Value
29	Fiscal Year Ends	Dec 30, 2020
30	Most Recent Quarter (mrq)	Sep 29, 2021

Profitability

Trading Information

Stock Price History

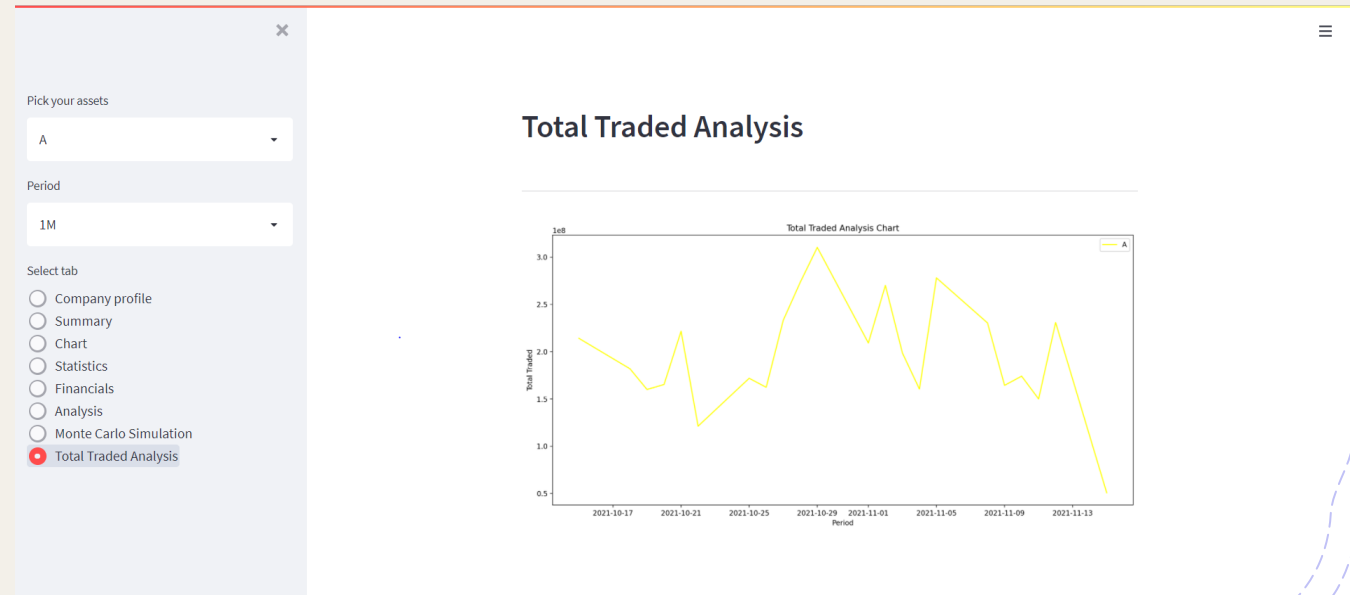
	0	1
0	Market Cap (intraday) 5	13.81B
1	Enterprise Value 3	45.26B
2	Trailing P/E	2.81
3	Forward P/E 1	1.11k
4	PEG Ratio (5 yr expected) 1	<NA>
5	Price/Sales (ttm)	0.52
6	Price/Book (mrq)	<NA>

Share Statistics

	Attribute	Value
7	Avg Vol (3 month) 3	29.54M
8	Avg Vol (10 day) 3	34.15M
9	Shares Outstanding 5	647.51M
10	Implied Shares Outstandin...	<NA>
11	Float 8	642.39M
12	PE Ratio (intraday) 1	0.00%

# Total Traded Analysis

- + This tab shows us the total traded price of each stock within a given period
- + It helps us analyse the trading volume for a stock. Where the volume is used as a unit of measurement for market's activity during a period of time
- + Reference of code  
<https://www.youtube.com/watch?v=57qAxRV577c&t=267s>



# Body of Code for Libraries

This code contains all the library functions that was used for the entire project

```
1 from os import stat
2 from attr import attrib
3 import streamlit as st
4 from streamlit.type_util import Key
5 import yfinance as yf
6 import pandas as pd
7 import yahoo_fin.stock_info as si
8 from datetime import datetime, timedelta
9 import plotly.graph_objects as go
10 from plotly.subplots import make_subplots
11 import matplotlib.pyplot as plt
12 import numpy as np
13 from pandas_datareader import data as pdr
14
15
```



# Body of Code for Time Frames

- + This contains the code for the interval periods for the stock trade
- + I also imported the relative delta library in order to compute for the dates
- + The current\_date is meant to fetch the date as today's date(present date)
- + Then the if-elif loop is meant to loop through the time intervals stated in the timeframe in months and display the data for the selected period

```
17 # main date function to compute for intervals
18 def date_function(n):
19     from dateutil.relativedelta import relativedelta
20     current_date = datetime.today()
21     print('Current Date: ', current_date)
22     timeframe = 0
23     if(n == '1M'):
24         timeframe = 1
25     elif(n == '3M'):
26         timeframe = 3
27     elif(n == '6M'):
28         timeframe = 6
29     elif(n == 'YTD'):
30         timeframe = datetime.now().month
31     elif(n == '1Y'):
32         timeframe = 12
33     elif(n == '3Y'):
34         timeframe = 36
35     elif(n == '5Y'):
36         timeframe = 60
37     past_date = current_date - relativedelta(months=timeframe)
38     # Convert datetime object to string in required format
39     date_format = '%Y/%m/%d'
40     past_date_str = past_date.strftime(date_format)
41     print('Date (as string) - 20 months before current date: ', past_date_str)
42     return past_date_str
43
```

# Body of Code for Selection Bars

- + The ticker list basically contains all the sp5000 tickers on the yahoo finance
- + Ticker item is called with the streamlit sidebar+selectbox function to choose the stock of choice on the menu sidebar
- + The range array consist of the time intervals in a list
- + The period is also assigned with a streamlit sidebar+select box for the time frames
- + The key helps to pair values with requirements that keys returned are unique

```
# set default ticker to sp500
ticker_list = ['-'] + si.tickers_sp500()
# create ticker items dropdown
ticker_item = st.sidebar.selectbox(
    'Pick your assets', ticker_list, key="menu_sidebar")

# duration of time
range_array = ['1M', '3M', '6M', 'YTD', '1Y', '3Y', '5Y', 'MAX']
# create sidebar dropdown for duration
period = st.sidebar.selectbox('Period', range_array, key="time-period-tab1")
```

# Body of Code for Tab 1

- + Defining tab1 function with the first few streamlit functions to give us the Title and Header of the page
- + The function GetCompanyInfo is a function that returns the company information of any given ticker
- + The get\_company\_info helps to fetch the company information from yahoo finance
- + The if loop holds the tickers and once a ticker is selected the information of the company will be displayed

```
56
57 def tab1():
58     # Add dashboard title and description
59     st.title("YAHOO FINANCIAL DASHBOARD")
60     st.write("Data source: Yahoo Finance")
61     st.header('Tab 1 - COMPANY PROFILE')
62
63     @st.cache
64     def GetCompanyInfo(ticker_item):
65         return si.get_company_info(ticker_item)
66
67     # show warning when stock isn't selected
68     if(ticker_item == '-'):
69         st.warning('Select a stock to see data')
70
71     if ticker_item != '-':
72         info = GetCompanyInfo(ticker_item)
73         info['Value'] = info['Value'].astype(str)
74         st.dataframe(info, height=2000)
75
```

```

def tab2():
    st.header('Summary')
    st.write('*****')
    # set sidebar header
    st.sidebar.header('Select Tickers to Get Started')

    # Add table to show stock data

    @st.cache
    def GetCompanyInfo(ticker_item):
        return si.get_quote_table(ticker_item)

    # show warning when stock isn't selected
    if(ticker_item == '-'):
        st.warning('Select a stock to see data')

    if ticker_item != '-':
        info = GetCompanyInfo(ticker_item)
        data = pd.DataFrame.from_dict(info, orient='index').astype(str)
        st.dataframe(data, height=2000)

    start = pd.to_datetime(date_function(period))
    today = pd.to_datetime('today')

    # stock chart
    df = yf.download(ticker_item, start, today)
    st.line_chart(df)

    # update button to download and update the data
    st.sidebar.write('Select an option to either download or update')

    options = ['-', 'update', 'download']
    selected_option = st.sidebar.selectbox('Select an action', options)

    # download dataframe as csv after adding filename (filename optional. it would result .de
    def download_as_csv(filename):
        if not filename:
            df.to_csv('default.csv', index=False)
        else:

```