Stock Analysis of top 5 companies of the world

****In this project I'll be using python programming, pandas, matplotlib and other libraries. I'll examine the stocks of top 5 Companies in the World on the basis of market cap. I'll use the yfinance library to extract data and then do operations to evaluate and visualise data.****

The companies I'll conside in this evaluation are: 1- Apple Inc 2- Microsoft 3- Saudi Aramco 4- Alphabet 5- Amazon

In [1]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\nishant\anaconda3\lib\site-pa
ckages (1.3.3)
Requirement already satisfied: numpy>=1.17.3 in c:\users\nishant\anaconda3\lib
\site-packages (from pandas) (1.20.3)
Requirement already satisfied: pytz>=2017.3 in c:\users\nishant\anaconda3\lib\s
ite-packages (from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\nishant\anaco
nda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\nishant\anaconda3\lib\site-
packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\nishant\anaconda3\lib\site-pac
kages (1.20.3)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\nishant\anaconda3\lib\sit
e-packages (3.4.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\nishant\anaconda3\lib
\site-packages (from matplotlib) (8.4.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\nishant\anaconda3\l
ib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\nishant\anaconda3
\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\nishant\anacond
a3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\nishant\anaconda3\lib\s
ite-packages (from matplotlib) (0.10.0)
Requirement already satisfied: numpy>=1.16 in c:\users\nishant\anaconda3\lib\si
te-packages (from matplotlib) (1.20.3)
Requirement already satisfied: six in c:\users\nishant\anaconda3\lib\site-packa
ges (from cycler>=0.10->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install yfinance
```

```
Requirement already satisfied: yfinance in c:\users\nishant\anaconda3\lib\site-
packages (0.1.67)
Requirement already satisfied: numpy>=1.15 in c:\users\nishant\anaconda3\lib\si
te-packages (from yfinance) (1.20.3)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\nishant\anaconda
3\lib\site-packages (from yfinance) (0.0.10)
Requirement already satisfied: requests>=2.20 in c:\users\nishant\anaconda3\lib
\site-packages (from yfinance) (2.26.0)
Requirement already satisfied: pandas>=0.24 in c:\users\nishant\anaconda3\lib\s
ite-packages (from yfinance) (1.3.3)
Requirement already satisfied: lxml>=4.5.1 in c:\users\nishant\anaconda3\lib\si
te-packages (from yfinance) (4.6.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\nishant\anaco
nda3\lib\site-packages (from pandas>=0.24->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\nishant\anaconda3\lib\s
ite-packages (from pandas>=0.24->yfinance) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\nishant\anaconda3\lib\site-
packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.16.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\nishant\anacon
da3\lib\site-packages (from requests>=2.20->yfinance) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\nishant\an
aconda3\lib\site-packages (from requests>=2.20->yfinance) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\nishant\anaconda3\lib\s
ite-packages (from requests>=2.20->yfinance) (3.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nishant\anaconda3
\lib\site-packages (from requests>=2.20->yfinance) (2021.10.8)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import yfinance as yf
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

Extracting Apple inc data using yfinace

In [6]:
```
apple = yf.Ticker("AAPL")
```

Using the ticker object and the function history extract stock information and save it in a dataframe named tesla_data. Set the period parameter to max so we get information for the maximum amount of time.

In [7]:
```
apple_data = apple.history(period="max")
```

In [8]:
```
apple_data.reset_index(inplace = True)
```

In [9]:
```
apple_data.tail(10)
```

Out[9]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
10431	2022-04-27	155.681266	159.555563	155.152045	156.340302	88063200	0.00	0.0
10432	2022-04-28	159.016359	164.278631	158.696821	163.399918	130216800	0.00	0.0
10433	2022-04-29	161.602556	165.956160	157.019294	157.418701	131587100	0.00	0.0
10434	2022-05-02	156.480090	157.997849	153.045135	157.728256	123055300	0.00	0.0
10435	2022-05-03	157.917967	160.474224	156.090665	159.246017	88966500	0.00	0.0
10436	2022-05-04	159.435739	166.235745	159.026336	165.776428	108256500	0.00	0.0
10437	2022-05-05	163.609623	163.839282	154.722671	156.540009	130525300	0.00	0.0
10438	2022-05-06	156.009995	159.440002	154.179993	157.279999	116055700	0.23	0.0
10439	2022-05-09	154.929993	155.830002	151.490005	152.059998	131577900	0.00	0.0
10440	2022-05-10	155.520004	156.740005	152.929993	154.509995	115276400	0.00	0.0

In [10]:
```
newdfapp = apple_data
```

dropping Stock Splits and dividends from all the data sources because it is better to remove the irrevelant data from analysis point of view.

In [11]:
```
newdfapp=newdfapp.drop('Stock Splits' ,axis = 1)
```

In [12]:
```
newdfapp=newdfapp.drop('Dividends' ,axis = 1)
```

In [13]:
```
newdfapp.tail(10)
```

Out[13]:

	Date	Open	High	Low	Close	Volume
10431	2022-04-27	155.681266	159.555563	155.152045	156.340302	88063200
10432	2022-04-28	159.016359	164.278631	158.696821	163.399918	130216800
10433	2022-04-29	161.602556	165.956160	157.019294	157.418701	131587100
10434	2022-05-02	156.480090	157.997849	153.045135	157.728256	123055300
10435	2022-05-03	157.917967	160.474224	156.090665	159.246017	88966500
10436	2022-05-04	159.435739	166.235745	159.026336	165.776428	108256500
10437	2022-05-05	163.609623	163.839282	154.722671	156.540009	130525300
10438	2022-05-06	156.009995	159.440002	154.179993	157.279999	116055700
10439	2022-05-09	154.929993	155.830002	151.490005	152.059998	131577900
10440	2022-05-10	155.520004	156.740005	152.929993	154.509995	115276400

Extracting Microsoft data using yfinace

In [14]:
```
microsoft = yf.Ticker("MSFT")
```

In [15]:
```
microsoft_data = microsoft.history(period="max")
```

In [16]:
```
microsoft_data.reset_index(inplace = True)
```

In [17]:
```
microsoft_data.tail(10)
```

Out[17]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
9105	2022-04-27	282.100006	290.970001	279.160004	283.220001	63477700	0.0	0.0
9106	2022-04-28	285.190002	290.980011	281.459991	289.630005	33646600	0.0	0.0
9107	2022-04-29	288.609985	289.880005	276.500000	277.519989	37025000	0.0	0.0
9108	2022-05-02	277.709991	284.940002	276.220001	284.470001	35151100	0.0	0.0
9109	2022-05-03	283.959991	284.130005	280.149994	281.779999	25978600	0.0	0.0
9110	2022-05-04	282.589996	290.880005	276.730011	289.980011	33599300	0.0	0.0
9111	2022-05-05	285.540009	286.350006	274.339996	277.350006	43260400	0.0	0.0
9112	2022-05-06	274.809998	279.250000	271.269989	274.730011	37748300	0.0	0.0
9113	2022-05-09	270.059998	272.359985	263.320007	264.579987	47726000	0.0	0.0
9114	2022-05-10	271.690002	273.750000	265.070007	269.500000	39292300	0.0	0.0

In [18]:
```
newdfmic = microsoft_data
```

In [19]:
```
newdfmic=newdfmic.drop('Stock Splits' ,axis = 1)
```

In [20]:
```
newdfmic=newdfmic.drop('Dividends' ,axis = 1)
```

In [21]:
```
newdfmic.tail(10)
```

Out[21]:

	Date	Open	High	Low	Close	Volume
9105	2022-04-27	282.100006	290.970001	279.160004	283.220001	63477700
9106	2022-04-28	285.190002	290.980011	281.459991	289.630005	33646600
9107	2022-04-29	288.609985	289.880005	276.500000	277.519989	37025000
9108	2022-05-02	277.709991	284.940002	276.220001	284.470001	35151100
9109	2022-05-03	283.959991	284.130005	280.149994	281.779999	25978600
9110	2022-05-04	282.589996	290.880005	276.730011	289.980011	33599300
9111	2022-05-05	285.540009	286.350006	274.339996	277.350006	43260400
9112	2022-05-06	274.809998	279.250000	271.269989	274.730011	37748300
9113	2022-05-09	270.059998	272.359985	263.320007	264.579987	47726000
9114	2022-05-10	271.690002	273.750000	265.070007	269.500000	39292300

Extracting Saudi Aramco data using yfinace

In [22]:
```
saudi_armanco = yf.Ticker("2222.SR")
```

In [23]:
```
saudi_armanco_data = saudi_armanco.history(period="max")
```

In [24]:
```
saudi_armanco_data.reset_index(inplace = True)
```

In [25]:
```
saudi_armanco_data.tail(10)
```

Out[25]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
594	2022-04-21	43.400002	43.549999	42.900002	43.000000	7413831	0.0	0.0
595	2022-04-24	42.900002	43.000000	42.750000	43.000000	2950269	0.0	0.0
596	2022-04-25	42.950001	43.000000	42.750000	42.900002	4537787	0.0	0.0
597	2022-04-26	43.000000	43.400002	42.900002	43.349998	7143194	0.0	0.0
598	2022-04-27	43.349998	44.700001	43.250000	44.349998	12967209	0.0	0.0
599	2022-04-28	44.750000	44.900002	44.450001	44.900002	13157964	0.0	0.0
600	2022-05-08	44.799999	46.099998	44.700001	45.849998	13940899	0.0	0.0
601	2022-05-09	45.849998	46.250000	45.700001	45.950001	12750884	0.0	0.0
602	2022-05-10	45.950001	46.200001	45.250000	45.950001	9559481	0.0	0.0
603	2022-05-11	45.799999	46.150002	45.750000	45.849998	3285808	0.0	0.0

In [26]:
```
newdfsau = saudi_armanco_data
```

In [27]:
```
newdfsau=newdfsau.drop('Stock Splits' ,axis = 1)
```

In [28]:
```
newdfsau=newdfsau.drop('Dividends' ,axis = 1)
```

In [29]:
```
newdfsau.tail(10)
```

Out[29]:

	Date	Open	High	Low	Close	Volume
594	2022-04-21	43.400002	43.549999	42.900002	43.000000	7413831
595	2022-04-24	42.900002	43.000000	42.750000	43.000000	2950269
596	2022-04-25	42.950001	43.000000	42.750000	42.900002	4537787
597	2022-04-26	43.000000	43.400002	42.900002	43.349998	7143194
598	2022-04-27	43.349998	44.700001	43.250000	44.349998	12967209
599	2022-04-28	44.750000	44.900002	44.450001	44.900002	13157964
600	2022-05-08	44.799999	46.099998	44.700001	45.849998	13940899
601	2022-05-09	45.849998	46.250000	45.700001	45.950001	12750884
602	2022-05-10	45.950001	46.200001	45.250000	45.950001	9559481
603	2022-05-11	45.799999	46.150002	45.750000	45.849998	3285808

Extracting Alphabet data using yfinace

In [30]:
```
alphabet = yf.Ticker("GOOG")
```

In [31]:
```
alphabet_data = alphabet.history(period="max")
```

In [32]:
```
alphabet_data.reset_index(inplace = True)
```

In [33]:
```
alphabet_data.tail(10)
```

Out[33]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
4453	2022-04-27	2287.459961	2350.000000	2262.485107	2300.409912	3111900	0	0.0
4454	2022-04-28	2342.300049	2408.770020	2302.877930	2388.229980	1839500	0	0.0
4455	2022-04-29	2351.560059	2379.199951	2293.879883	2299.330078	1683500	0	0.0
4456	2022-05-02	2278.129883	2346.790039	2267.989990	2343.139893	1514000	0	0.0
4457	2022-05-03	2335.300049	2386.000000	2332.540039	2362.590088	1060800	0	0.0
4458	2022-05-04	2360.070068	2462.860107	2314.770020	2451.500000	1661600	0	0.0
4459	2022-05-05	2404.409912	2424.665039	2303.649902	2334.929932	2154500	0	0.0
4460	2022-05-06	2310.379883	2349.969971	2282.860107	2313.199951	1764000	0	0.0
4461	2022-05-09	2266.070068	2311.258057	2251.030029	2261.679932	1726000	0	0.0
4462	2022-05-10	2320.810059	2333.820068	2267.666016	2291.689941	1557000	0	0.0

In [34]:
```
newdfalp = alphabet_data
```

In [35]:
```
newdfalp=newdfalp.drop('Stock Splits' ,axis = 1)
```

In [36]:
```
newdfalp=newdfalp.drop('Dividends' ,axis = 1)
```

In [37]:
```
newdfalp.tail(10)
```

Out[37]:

	Date	Open	High	Low	Close	Volume
4453	2022-04-27	2287.459961	2350.000000	2262.485107	2300.409912	3111900
4454	2022-04-28	2342.300049	2408.770020	2302.877930	2388.229980	1839500
4455	2022-04-29	2351.560059	2379.199951	2293.879883	2299.330078	1683500
4456	2022-05-02	2278.129883	2346.790039	2267.989990	2343.139893	1514000
4457	2022-05-03	2335.300049	2386.000000	2332.540039	2362.590088	1060800
4458	2022-05-04	2360.070068	2462.860107	2314.770020	2451.500000	1661600
4459	2022-05-05	2404.409912	2424.665039	2303.649902	2334.929932	2154500
4460	2022-05-06	2310.379883	2349.969971	2282.860107	2313.199951	1764000
4461	2022-05-09	2266.070068	2311.258057	2251.030029	2261.679932	1726000
4462	2022-05-10	2320.810059	2333.820068	2267.666016	2291.689941	1557000

Extracting amazon data using yfinace

In [38]:
```
amazon = yf.Ticker("AMZN")
```

In [39]:
```
amazon_data = amazon.history(period="max")
```

In [40]:
```
amazon_data.reset_index(inplace = True)
```

In [41]:
```
amazon_data.tail(10)
```

Out[41]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
6279	2022-04-27	2803.830078	2838.969971	2715.669922	2763.340088	3566800	0	0.0
6280	2022-04-28	2843.560059	2918.750000	2806.000000	2891.929932	5865800	0	0.0
6281	2022-04-29	2596.979980	2615.219971	2432.500000	2485.629883	13616500	0	0.0
6282	2022-05-02	2448.020020	2493.360107	2367.500000	2490.000000	7439400	0	0.0
6283	2022-05-03	2481.070068	2524.409912	2456.500000	2485.070068	3956700	0	0.0
6284	2022-05-04	2472.000000	2520.000000	2383.659912	2518.570068	5537300	0	0.0
6285	2022-05-05	2460.000000	2469.979980	2301.449951	2328.139893	7219600	0	0.0
6286	2022-05-06	2297.000000	2381.010010	2261.629883	2295.449951	6206700	0	0.0
6287	2022-05-09	2226.250000	2280.000000	2159.139893	2175.780029	6406200	0	0.0
6288	2022-05-10	2225.000000	2252.850098	2143.419922	2177.179932	5264500	0	0.0

In [42]:
```
print(type(amazon_data))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [43]:
```
amazon_data=amazon_data.drop('Stock Splits' ,axis = 1)
```

In [44]:
```
amazon_data=amazon_data.drop('Dividends' ,axis = 1)
```

In [45]:
```
amazon_data.tail(10)
```

Out[45]:

	Date	Open	High	Low	Close	Volume
6279	2022-04-27	2803.830078	2838.969971	2715.669922	2763.340088	3566800
6280	2022-04-28	2843.560059	2918.750000	2806.000000	2891.929932	5865800
6281	2022-04-29	2596.979980	2615.219971	2432.500000	2485.629883	13616500
6282	2022-05-02	2448.020020	2493.360107	2367.500000	2490.000000	7439400
6283	2022-05-03	2481.070068	2524.409912	2456.500000	2485.070068	3956700
6284	2022-05-04	2472.000000	2520.000000	2383.659912	2518.570068	5537300
6285	2022-05-05	2460.000000	2469.979980	2301.449951	2328.139893	7219600
6286	2022-05-06	2297.000000	2381.010010	2261.629883	2295.449951	6206700
6287	2022-05-09	2226.250000	2280.000000	2159.139893	2175.780029	6406200
6288	2022-05-10	2225.000000	2252.850098	2143.419922	2177.179932	5264500

In [46]:
```
from matplotlib import pyplot as plt
```

In [47]:
```
get_ipython().run_line_magic('matplotlib', 'inline')
```

Downloading all the data from 01-01-2021 to 11-05-2022

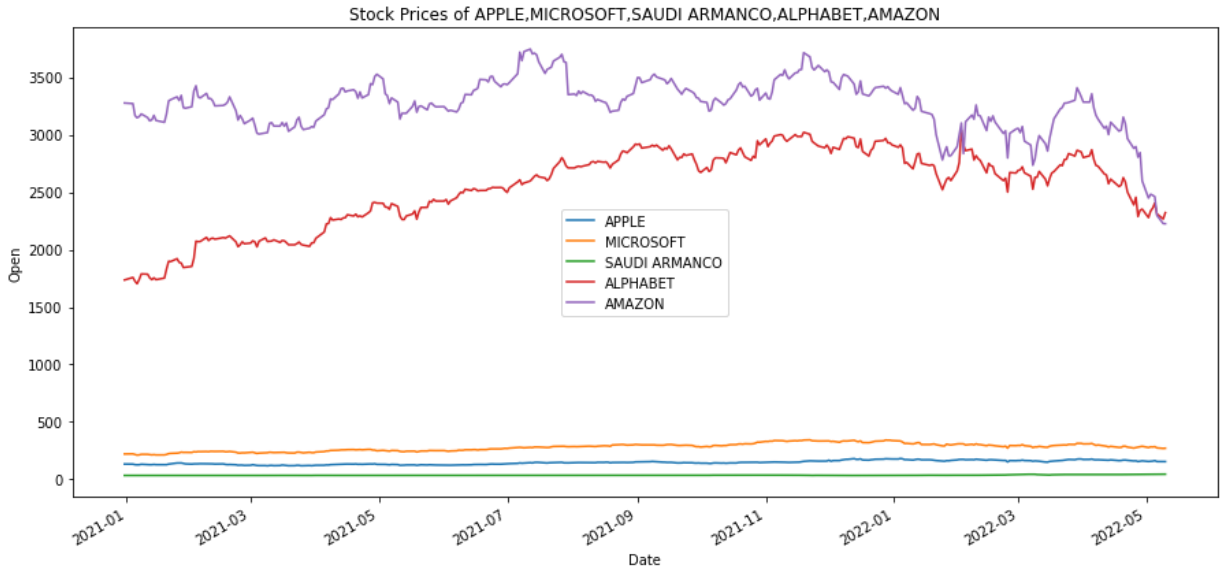
In [49]:
```
start = "2021-01-01"
end = '2022-05-11'
apple = yf.download('AAPL',start,end)
microsoft = yf.download('MSFT',start,end)
saudi_armanco = yf.download('2222.SR',start,end)
alphabet = yf.download('GOOG',start,end)
amazon = yf.download('AMZN',start,end)
```

```
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
```

Now let's analyse all the companies with criteria OPEN

In [53]:
```
apple['Open'].plot(label = 'APPLE', figsize = (15,7))
microsoft['Open'].plot(label = "MICROSOFT")
saudi_armanco['Open'].plot(label = 'SAUDI ARMANCO')
alphabet['Open'].plot(label = "ALPHABET")
amazon['Open'].plot(label = "AMAZON")
plt.title('Stock Prices of APPLE,MICROSOFT,SAUDI ARMANCO,ALPHABET and AMAZON')
plt.ylabel("Open")
plt.legend()
```

Out[53]: <matplotlib.legend.Legend at 0x2963e92eb80>

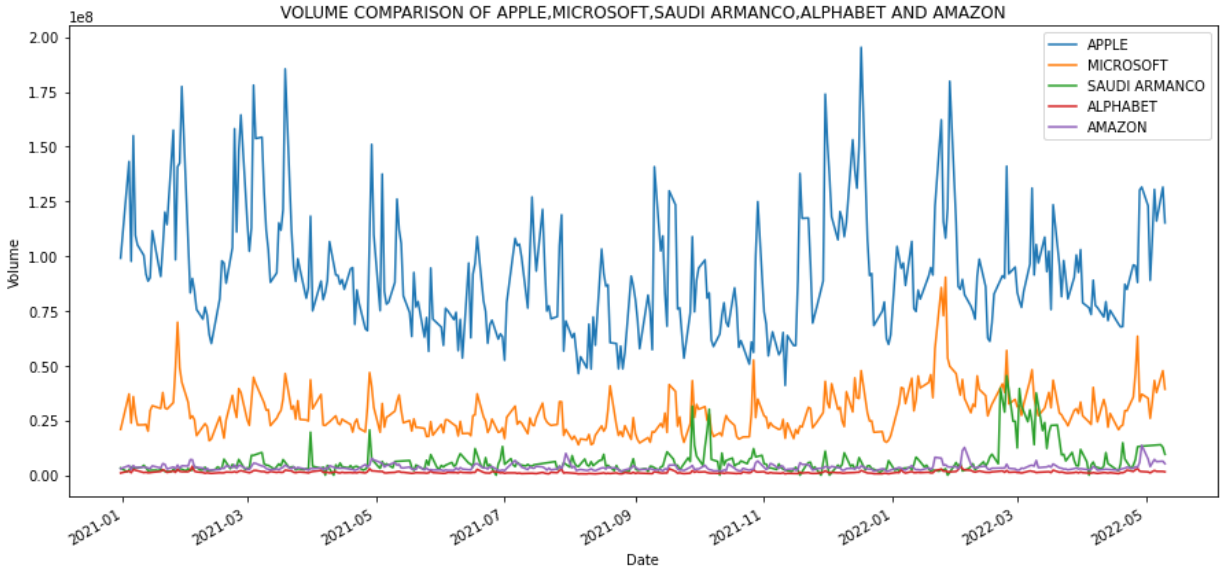


Through this we can conclude that the prices of amazon is higher than the others through out the duration considered. Also, it can be concluded that they is not much increase in the prices of saudi armanco,microsoft and apple.

Now let's analyse the volume traded of the companies

In [54]:
```
apple['Volume'].plot(label = 'APPLE', figsize = (15,7))
microsoft['Volume'].plot(label = "MICROSOFT")
saudi_armanco['Volume'].plot(label = 'SAUDI ARMANCO')
alphabet['Volume'].plot(label = "ALPHABET")
amazon['Volume'].plot(label = "AMAZON")
plt.title('Volume comparison of  APPLE,MICROSOFT,SAUDI ARMANCO,ALPHABET and AMAZO
plt.ylabel("Volume")
plt.legend()
```

Out[54]: <matplotlib.legend.Legend at 0x2964069c9a0>

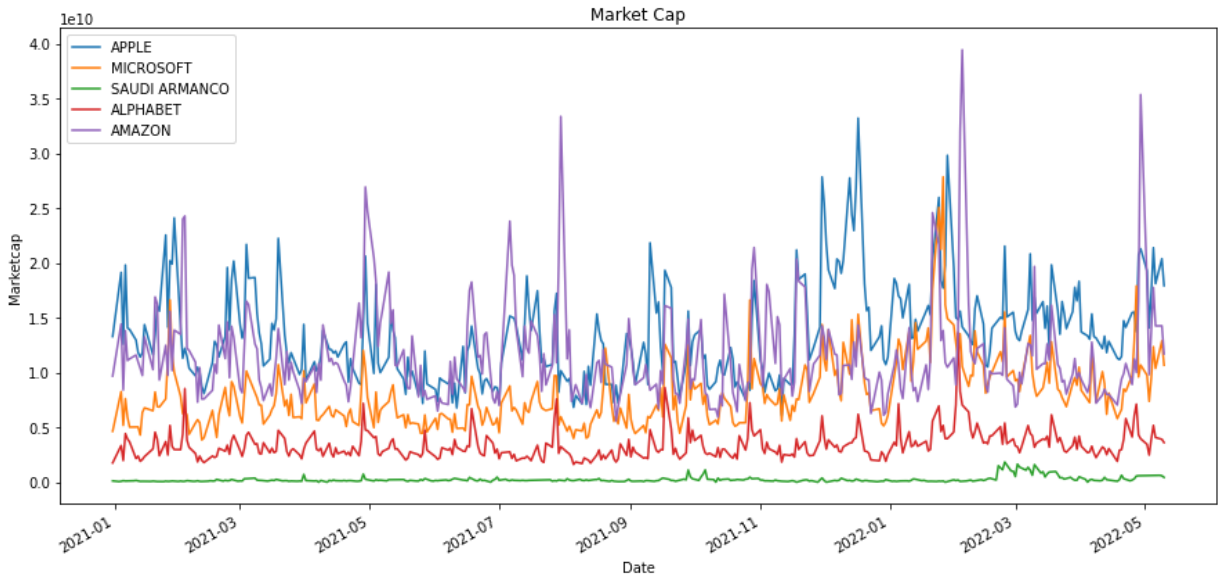


****Through the graph formed, it can be concluded that the stocks of APPLE are traded more when compared to the other four companies. Also, the trade of stock of alphabet is nearly constant.**

Now let's analyse the marketcap holded by all these companies which can be caluculated by multiplying Open and Volume records

In [56]:
```
apple['MarktCap'] = apple['Open'] * apple['Volume']
microsoft['MarktCap'] = microsoft['Open'] * microsoft['Volume']
saudi_armanco['MarktCap'] = saudi_armanco['Open'] * saudi_armanco['Volume']
alphabet['MarktCap'] = alphabet['Open'] * alphabet['Volume']
amazon['MarktCap'] = amazon['Open'] * amazon['Volume']
apple['MarktCap'].plot(label = 'APPLE', figsize = (15,7))
microsoft['MarktCap'].plot(label = 'MICROSOFT')
saudi_armanco['MarktCap'].plot(label = 'SAUDI ARMANCO')
alphabet['MarktCap'].plot(label = 'ALPHABET')
amazon['MarktCap'].plot(label = 'AMAZON')
plt.title('Market Cap')
plt.ylabel("Marketcap")
plt.legend()
```

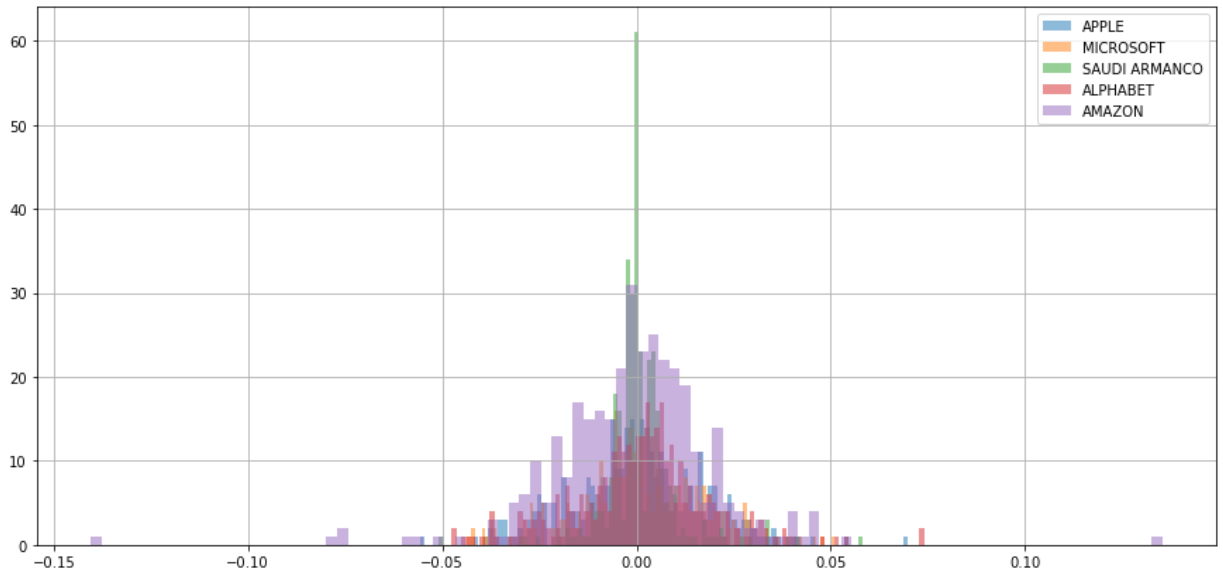
Out[56]:
```
<matplotlib.legend.Legend at 0x29640e7fb80>
```



Now let's check the volatality of the stocks. Volatality is the rate at which the price of the stock increases or decreases.

In [57]:
```
apple['returns'] = (apple['Close']/apple['Close'].shift(1)) -1
microsoft['returns'] = (microsoft['Close']/microsoft['Close'].shift(1))-1
saudi_armanco['returns'] = (saudi_armanco['Close']/saudi_armanco['Close'].shift(1
alphabet['returns'] = (alphabet['Close']/alphabet['Close'].shift(1)) -1
amazon['returns'] = (amazon['Close']/amazon['Close'].shift(1)) -1
apple['returns'].hist(bins = 100, label = 'APPLE', alpha = 0.5, figsize = (15,7))
microsoft['returns'].hist(bins = 100, label = 'MICROSOFT', alpha = 0.5)
saudi_armanco['returns'].hist(bins = 100, label = 'SAUDI ARMANCO', alpha = 0.5)
alphabet['returns'].hist(bins = 100, label = 'ALPHABET', alpha = 0.5)
amazon['returns'].hist(bins = 100, label = 'AMAZON', alpha = 0.5)
plt.legend()
```

Out[57]:
```
<matplotlib.legend.Legend at 0x29641367400>
```



It is clear that the volatility of amazon is slightly higher than the others

CONCLUSION - In this project, various aspects of the top 5 companies in the world were discussed and I also extracted the numeric data stats of all the companies taken into consideration.

LINKEDIN LINK - https://www.linkedin.com/in/nishant-gaurav-4b2753230/ (https://www.linkedin.com/in/nishant-gaurav-4b2753230/)