In [2]:
```
import pandas as pd
import yfinance as yf
from datetime import datetime
import plotly.express as px
```

In [3]:
```
data=pd.read_csv('C:/Users/VIP/Downloads/stocks.csv')
data
```

Out[3]:

	Ticker	Date	Open	High	Low	Close	Adj Close	Volume
0	AAPL	2023-02-07	150.639999	155.229996	150.639999	154.649994	154.414230	83322600
1	AAPL	2023-02-08	153.880005	154.580002	151.169998	151.919998	151.688400	64120100
2	AAPL	2023-02-09	153.779999	154.330002	150.419998	150.869995	150.639999	56007100
3	AAPL	2023-02-10	149.460007	151.339996	149.220001	151.009995	151.009995	57450700
4	AAPL	2023-02-13	150.949997	154.259995	150.919998	153.850006	153.850006	62199000
...	...	...	...	...	...	...	...	...
243	GOOG	2023-05-01	107.720001	108.680000	107.500000	107.709999	107.709999	20926300
244	GOOG	2023-05-02	107.660004	107.730003	104.500000	105.980003	105.980003	20343100
245	GOOG	2023-05-03	106.220001	108.129997	105.620003	106.120003	106.120003	17116300
246	GOOG	2023-05-04	106.160004	106.300003	104.699997	105.209999	105.209999	19780600
247	GOOG	2023-05-05	105.320000	106.440002	104.738998	106.214996	106.214996	20705300

248 rows × 8 columns

In [4]:
```
start_date = datetime.now() - pd.DateOffset(months=3)
end_date = datetime.now()
```

In [6]:
```
data['Ticker'].value_counts()
```

Out[6]:
```
Ticker
AAPL    62
MSFT    62
NFLX    62
GOOG    62
Name: count, dtype: int64
```

In [7]:
```
tickers = ['AAPL', 'MSFT', 'NFLX', 'GOOG']
df_list = []
for ticker in tickers:
    data = yf.download(ticker, start=start_date, end=end_date)
    df_list.append(data)
```

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

In [8]:
```
df = pd.concat(df_list, keys=tickers, names=['Ticker', 'Date'])
print(df.head())
```

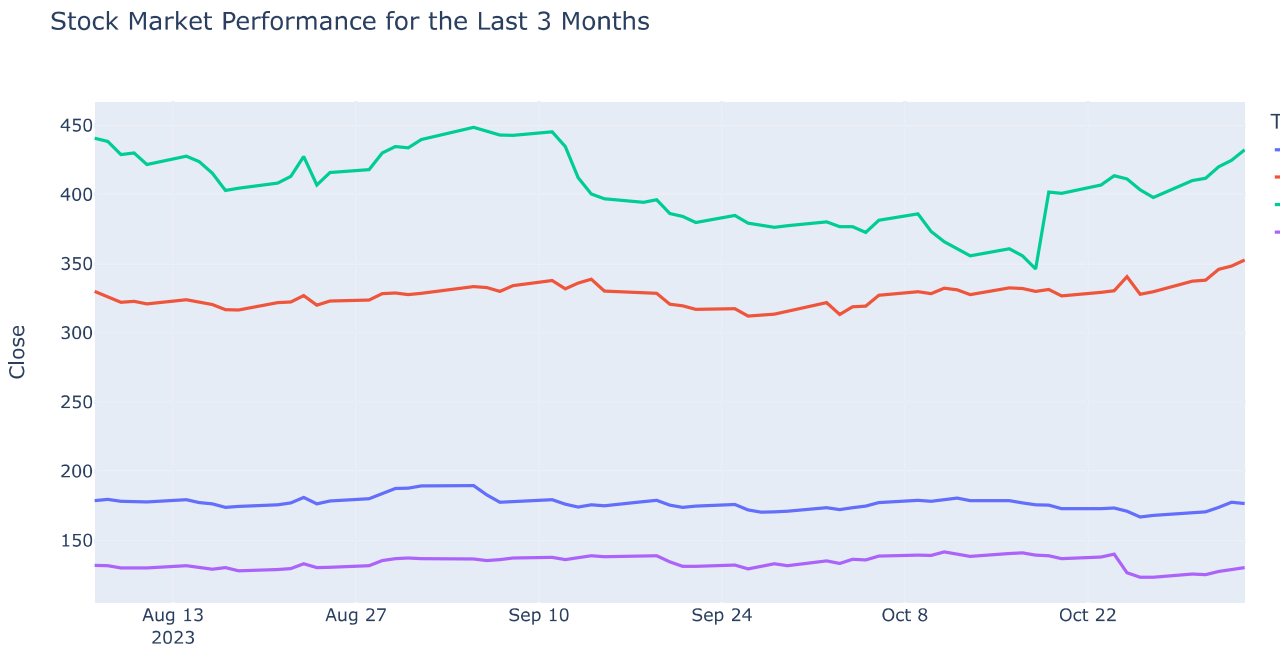
```
                          Open        High         Low       Close   Adj Close  \
Ticker Date
AAPL   2023-08-07  182.130005  183.130005  177.350006  178.850006  178.608810
       2023-08-08  179.690002  180.270004  177.580002  179.800003  179.557526
       2023-08-09  180.869995  180.929993  177.009995  178.190002  177.949707
       2023-08-10  179.479996  180.750000  177.600006  177.970001  177.729996
       2023-08-11  177.320007  178.619995  176.550003  177.789993  177.789993

                      Volume
Ticker Date
AAPL   2023-08-07  97576100
       2023-08-08  67823000
       2023-08-09  60378500
       2023-08-10  54686900
       2023-08-11  51988100
```

In [9]:
```
df = df.reset_index()
print(df.head())
```

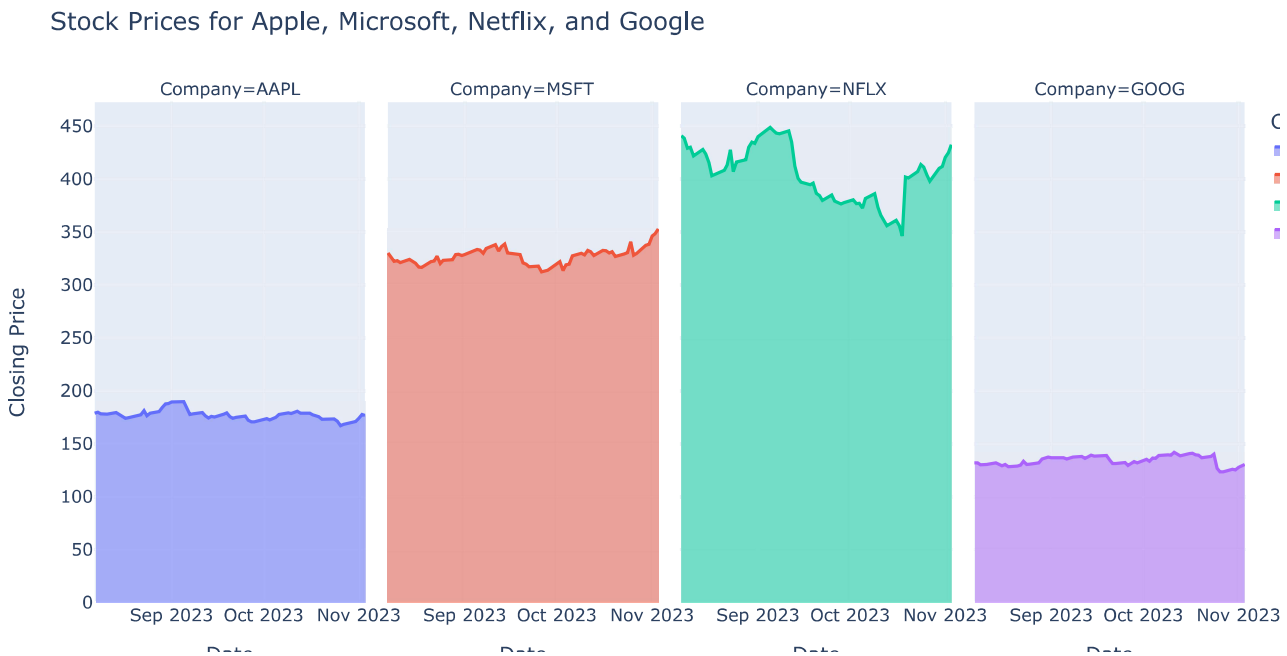
```
  Ticker       Date        Open        High         Low       Close  \
0   AAPL 2023-08-07  182.130005  183.130005  177.350006  178.850006
1   AAPL 2023-08-08  179.690002  180.270004  177.580002  179.800003
2   AAPL 2023-08-09  180.869995  180.929993  177.009995  178.190002
3   AAPL 2023-08-10  179.479996  180.750000  177.600006  177.970001
4   AAPL 2023-08-11  177.320007  178.619995  176.550003  177.789993

    Adj Close    Volume
0  178.608810  97576100
1  179.557526  67823000
2  177.949707  60378500
3  177.729996  54686900
4  177.789993  51988100
```

Now let’s have a look at the performance in the stock market of all the companies:

In [10]:
```
fig = px.line(df, x='Date',
              y='Close',
              color='Ticker',
              title="Stock Market Performance for the Last 3 Months")
fig.show()
```



In [11]:
```
fig = px.area(df, x='Date', y='Close', color='Ticker',
              facet_col='Ticker',
              labels={'Date':'Date', 'Close':'Closing Price', 'Ticker':'Company'},
              title='Stock Prices for Apple, Microsoft, Netflix, and Google')
fig.show()
```



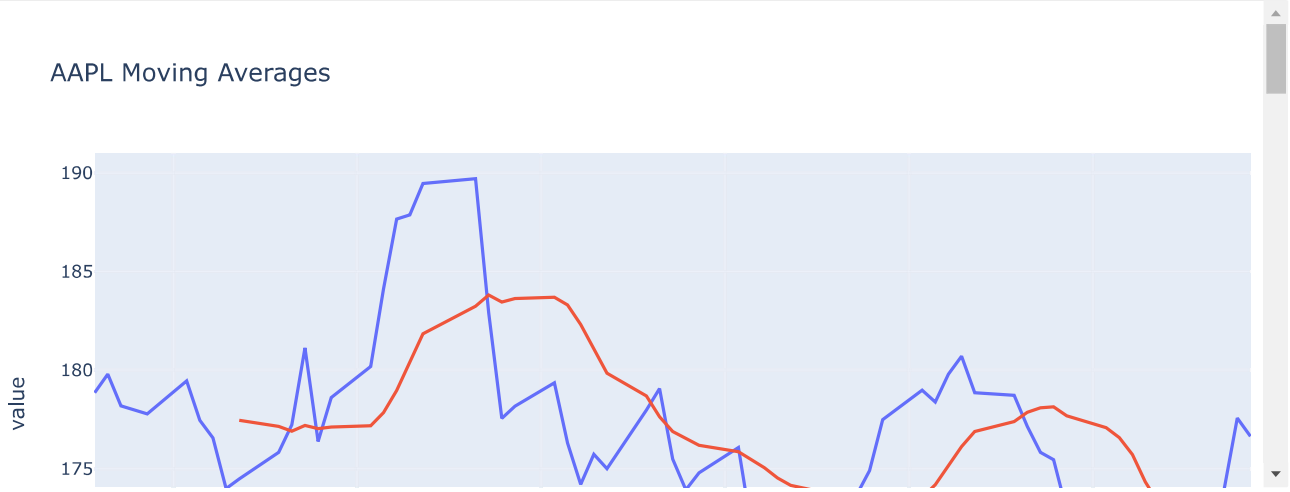
 # analyze moving averages, which provide a useful way to identify trends and patterns in each company's stock price movements over a period of time:

 # to calculate the 10-day moving average of a time series, you can use rolling(window=10)

In [12]:
```
df['MA10'] = df.groupby('Ticker')['Close'].rolling(window=10).mean().reset_index(0, drop=True)
for ticker, group in df.groupby('Ticker'):
    print(f'Moving Averages for {ticker}')
    print(group[['MA10']])
```

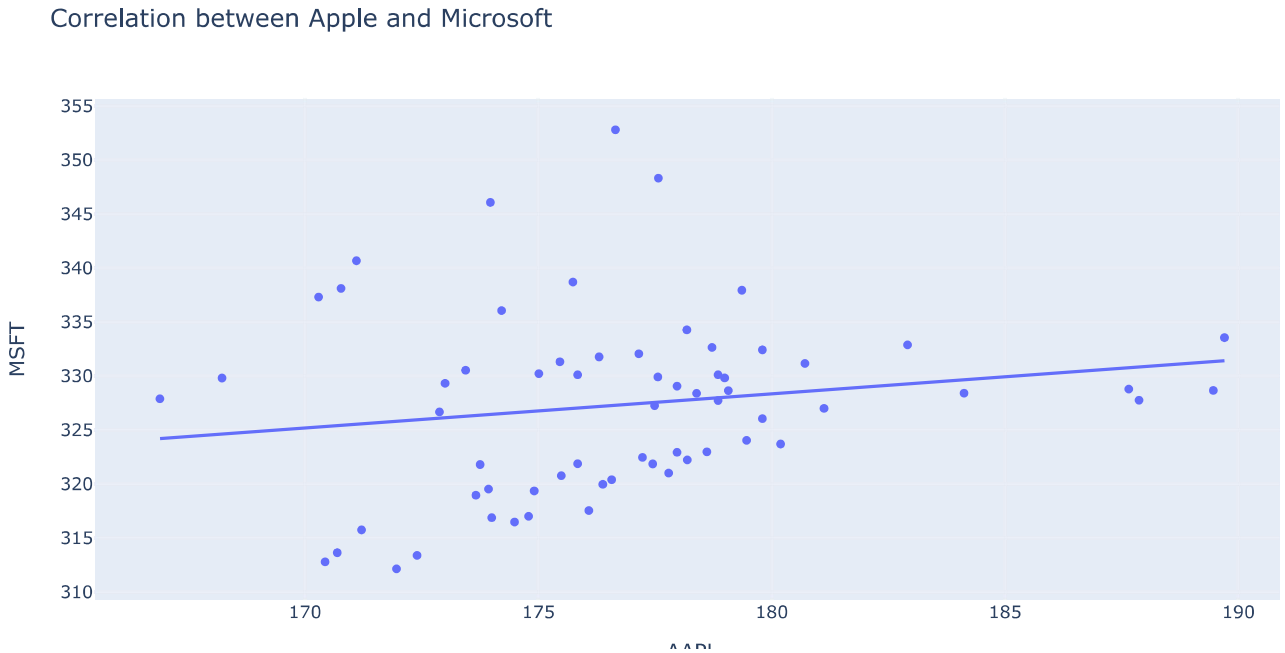
```
Moving Averages for AAPL
          MA10
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
..         ...
59  172.427000
60  171.789001
61  171.602002
62  171.813002
63  172.190001

[64 rows x 1 columns]
Moving Averages for GOOG
           MA10
192         NaN
193         NaN
194         NaN
195         NaN
196         NaN
..          ...
251  133.327000
252  131.757999
253  130.587000
254  129.547000
255  128.909999

[64 rows x 1 columns]
Moving Averages for MSFT
           MA10
64          NaN
65          NaN
66          NaN
67          NaN
68          NaN
..          ...
123  331.569003
124  332.174002
125  333.770004
126  335.470004
127  338.083002

[64 rows x 1 columns]
Moving Averages for NFLX
           MA10
128         NaN
129         NaN
130         NaN
131         NaN
132         NaN
..          ...
187  394.794998
188  400.391998
189  407.791998
190  410.085999
191  413.225998

[64 rows x 1 columns]
```

In [13]:
```
for ticker, group in df.groupby('Ticker'):
    fig = px.line(group, x='Date', y=['Close', 'MA10'],
                  title=f"{ticker} Moving Averages")
    fig.show()
```



 # There is a strong linear relationship between the stock prices of Apple and Microsoft, which means that when the stock price of Apple increases, the stock price of Microsoft also increase

In [14]:
```
apple = df.loc[df['Ticker'] == 'AAPL', ['Date', 'Close']].rename(columns={'Close': 'AAPL'})
microsoft = df.loc[df['Ticker'] == 'MSFT', ['Date', 'Close']].rename(columns={'Close': 'MSFT'})
df_corr = pd.merge(apple, microsoft, on='Date')

# create a scatter plot to visualize the correlation
fig = px.scatter(df_corr, x='AAPL', y='MSFT',
                 trendline='ols',
                 title='Correlation between Apple and Microsoft')
fig.show()
```



In [ ]: