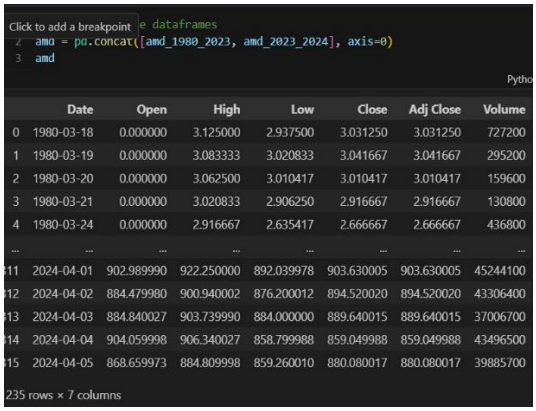
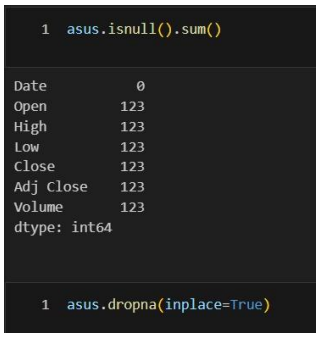
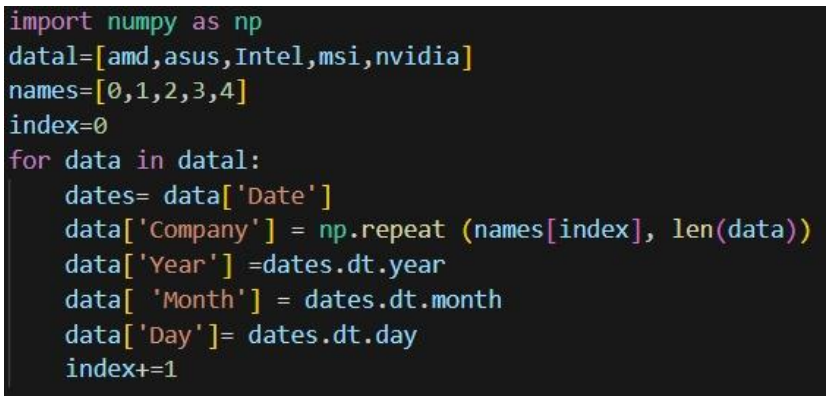
Data Collection and Preprocessing Phase

Date	15 April 2024
Team ID	Team - 738203
Project Title	Share Price Estimation Of TOP 5 GPU Companies
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Section	Description
Data Overview	In description we applied df.describe and we have counted number of columns, mean, min, max, standard deviation Shape of datasets amd (11235, 7) asus (6167, 7) Intel (11235, 7) msi (298, 7) nvidia (6470, 7) For all datasets there are 7 columns which are date, open, high, low, close, adj close, volume and there datatypes are object, float and integers
Univariate Analysis	N/A
Bivariate Analysis	Code dynamically creates a grid of subplots based on the number of columns in the df_plot DataFrame, plots each time series on a separate subplot using Seaborn's lineplot(), and configures the appearance of the plot for readability and aesthetics. It's a concise and effective way to visualize multiple time series of stock data simultaneously.
Multivariate Analysis	N/A
Outliers and Anomalies	In the datasets of asus there are 123 outliers and we have handled the null values with the help of dropna function

Data Preprocessing Code Screenshots	
Loading Data	
Handling Missing Data	
Data Transformation	N/A
Feature Engineering	
Save Processed Data	

```
Click to add a breakpoint e dataframes
amd = pd.concat([amd_1980_2023, amd_2023_2024], axis=0)
amd
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	1980-03-18	0.000000	3.125000	2.937500	3.031250	3.031250	727200
1	1980-03-19	0.000000	3.083333	3.020833	3.041667	3.041667	295200
2	1980-03-20	0.000000	3.062500	3.010417	3.010417	3.010417	159600
3	1980-03-21	0.000000	3.020833	2.906250	2.916667	2.916667	130800
4	1980-03-24	0.000000	2.916667	2.635417	2.666667	2.666667	436800
...	...	...	...	...	...	...	...
11	2024-04-01	902.989990	922.250000	892.039978	903.630005	903.630005	45244100
12	2024-04-02	884.479980	900.940002	876.200012	894.520020	894.520020	43306400
13	2024-04-03	884.840027	903.739990	884.000000	889.640015	889.640015	37006700
14	2024-04-04	904.059998	906.340027	858.799988	859.049988	859.049988	43496500
15	2024-04-05	868.659973	884.809998	859.260010	880.080017	880.080017	39885700

235 rows × 7 columns

```
asus.isnull().sum()
```

```
Date          0
Open        123
High        123
Low         123
Close       123
Adj Close   123
Volume      123
dtype: int64
```

```
asus.dropna(inplace=True)
```

```
# Changing dates to timestamps
datal=[amd,asus,Intel,msi,nvidia]
for data in datal:
    data['Date']=pd.to_datetime(data['Date'])
```

```
import numpy as np
datal=[amd,asus,Intel,msi,nvidia]
names=[0,1,2,3,4]
index=0
for data in datal:
    dates= data['Date']
    data['Company'] = np.repeat (names[index], len(data))
    data['Year'] =dates.dt.year
    data[ 'Month'] = dates.dt.month
    data['Day']= dates.dt.day
    index+=1
```