In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
dfdata=pd.read_csv(r"E:/asy notes/09 June/HCLTECH.csv")
```

In [4]:

```
dfdata
```

Out[4]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	
0	2000-01-11	HCLTECH	EQ	580.00	1550.0	1725.00	1492.00	1560.00	1554.45	1582.72	
1	2000-01-12	HCLTECH	EQ	1554.45	1560.0	1678.85	1560.00	1678.85	1678.85	1657.05	
2	2000-01-13	HCLTECH	EQ	1678.85	1790.0	1813.20	1781.00	1813.20	1813.20	1804.69	
3	2000-01-14	HCLTECH	EQ	1813.20	1958.3	1958.30	1835.00	1958.30	1958.30	1939.90	
4	2000-01-17	HCLTECH	EQ	1958.30	2115.0	2115.00	1801.65	1801.65	1801.65	1990.55	
...	...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.0	954.50	923.05	930.00	928.80	931.70	1
5296	2021-04-27	HCLTECH	EQ	928.80	931.0	938.55	923.40	930.30	928.85	928.06	
5297	2021-04-28	HCLTECH	EQ	928.85	931.2	935.85	921.75	925.90	923.80	926.63	
5298	2021-04-29	HCLTECH	EQ	923.80	929.7	929.70	907.10	910.30	909.55	914.34	
5299	2021-04-30	HCLTECH	EQ	909.55	905.0	915.00	895.40	900.10	898.95	904.98	1

5300 rows × 15 columns

In [5]:

```
dfdata["Date"]=pd.to_datetime(dfdata["Date"])
```

In [6]:

```
dfdata["Date"]
```

Out[6]:

```
0       2000-01-11
1       2000-01-12
2       2000-01-13
3       2000-01-14
4       2000-01-17
           ...
5295    2021-04-26
5296    2021-04-27
5297    2021-04-28
5298    2021-04-29
5299    2021-04-30
Name: Date, Length: 5300, dtype: datetime64[ns]
```

In [7]:

```
dfdata
```

Out[7]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	
0	2000-01-11	HCLTECH	EQ	580.00	1550.0	1725.00	1492.00	1560.00	1554.45	1582.72	
1	2000-01-12	HCLTECH	EQ	1554.45	1560.0	1678.85	1560.00	1678.85	1678.85	1657.05	
2	2000-01-13	HCLTECH	EQ	1678.85	1790.0	1813.20	1781.00	1813.20	1813.20	1804.69	
3	2000-01-14	HCLTECH	EQ	1813.20	1958.3	1958.30	1835.00	1958.30	1958.30	1939.90	
4	2000-01-17	HCLTECH	EQ	1958.30	2115.0	2115.00	1801.65	1801.65	1801.65	1990.55	
...	...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.0	954.50	923.05	930.00	928.80	931.70	1
5296	2021-04-27	HCLTECH	EQ	928.80	931.0	938.55	923.40	930.30	928.85	928.06	
5297	2021-04-28	HCLTECH	EQ	928.85	931.2	935.85	921.75	925.90	923.80	926.63	
5298	2021-04-29	HCLTECH	EQ	923.80	929.7	929.70	907.10	910.30	909.55	914.34	
5299	2021-04-30	HCLTECH	EQ	909.55	905.0	915.00	895.40	900.10	898.95	904.98	1

5300 rows × 15 columns

In [9]:

```
dfdata=dfdata.set_index("Date")
```

In [10]:

```
dfdata
```

Out[10]:

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volum
2000-01-11	HCLTECH	EQ	580.00	1550.0	1725.00	1492.00	1560.00	1554.45	1582.72	119220
2000-01-12	HCLTECH	EQ	1554.45	1560.0	1678.85	1560.00	1678.85	1678.85	1657.05	34485
2000-01-13	HCLTECH	EQ	1678.85	1790.0	1813.20	1781.00	1813.20	1813.20	1804.69	5300
2000-01-14	HCLTECH	EQ	1813.20	1958.3	1958.30	1835.00	1958.30	1958.30	1939.90	27095
2000-01-17	HCLTECH	EQ	1958.30	2115.0	2115.00	1801.65	1801.65	1801.65	1990.55	42880
...	...	...	...	...	...	...	...	...	...	.
2021-04-26	HCLTECH	EQ	955.65	940.0	954.50	923.05	930.00	928.80	931.70	1961997
2021-04-27	HCLTECH	EQ	928.80	931.0	938.55	923.40	930.30	928.85	928.06	640682
2021-04-28	HCLTECH	EQ	928.85	931.2	935.85	921.75	925.90	923.80	926.63	684567
2021-04-29	HCLTECH	EQ	923.80	929.7	929.70	907.10	910.30	909.55	914.34	858873
2021-04-30	HCLTECH	EQ	909.55	905.0	915.00	895.40	900.10	898.95	904.98	1092166

5300 rows × 14 columns

In [12]:

```
dfstock=dfdata[["Prev Close"]]
```

In [13]:

```
dfstock
```

Out[13]:

Date	Prev Close
2000-01-11	580.00
2000-01-12	1554.45
2000-01-13	1678.85
2000-01-14	1813.20
2000-01-17	1958.30
...	...
2021-04-26	955.65
2021-04-27	928.80
2021-04-28	928.85
2021-04-29	923.80
2021-04-30	909.55

5300 rows × 1 columns

In [14]:

```
plt.plot(dfstock)
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x284096a9fd0>]
```

In [16]:

```
dfstock=dfstock["2016-01-01":"2019-01-01"]
```

In [17]:

```
dfstock
```

Out[17]:

Date	Prev Close
2016-01-01	855.10
2016-01-04	845.85
2016-01-05	845.95
2016-01-06	842.80
2016-01-07	841.40
...	...
2018-12-26	937.85
2018-12-27	942.80
2018-12-28	948.80
2018-12-31	958.35
2019-01-01	964.35

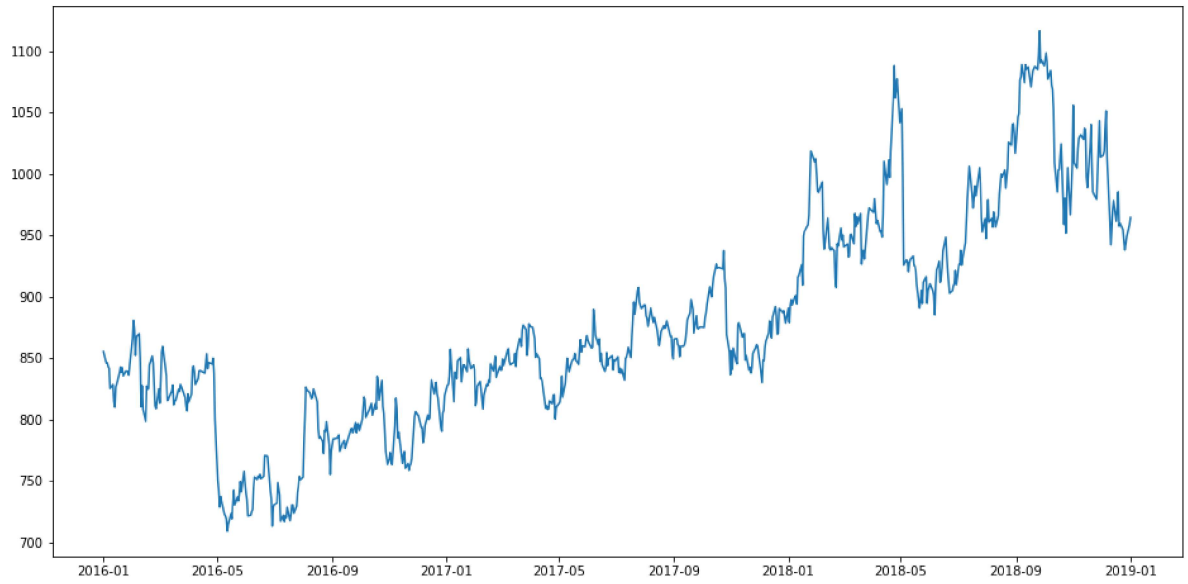
742 rows × 1 columns

In [18]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock)
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x28409ec2eb0>]
```



In [21]:

```
rollmean=dfstock.rolling(12).mean()
```

In [22]:

```
rollstd=dfstock.rolling(12).std()
```

In [23]:

```
rollmean
```

Out[23]:

Date	Prev Close
2016-01-01	NaN
2016-01-04	NaN
2016-01-05	NaN
2016-01-06	NaN
2016-01-07	NaN
...	...
2018-12-26	961.425000
2018-12-27	959.829167
2018-12-28	960.395833
2018-12-31	960.520833
2019-01-01	959.945833

742 rows × 1 columns

In [24]:

```
rollstd
```

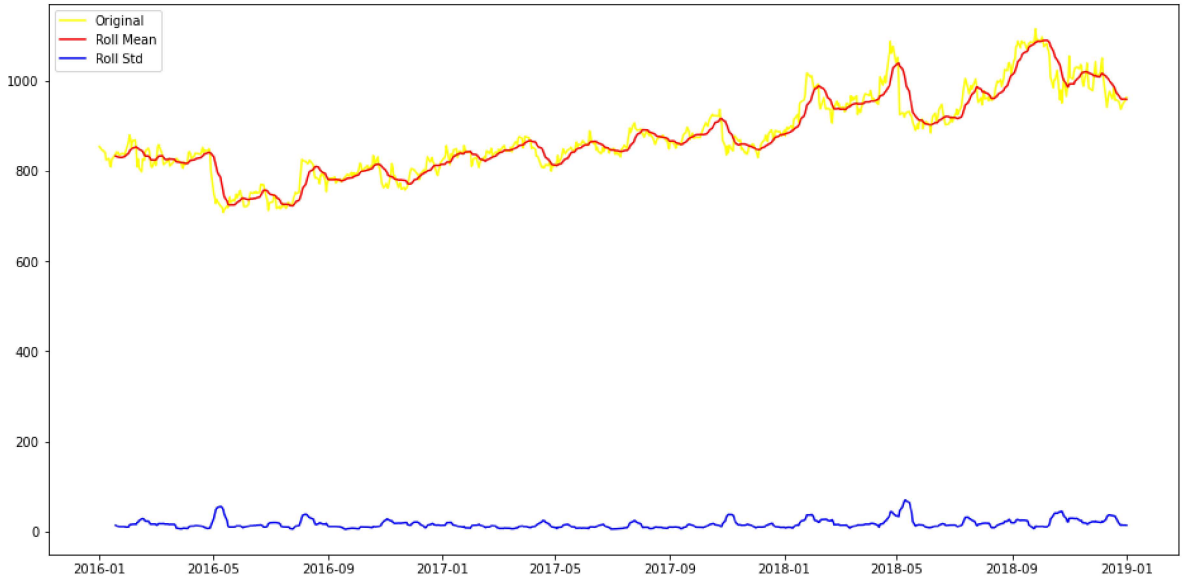
Out[24]:

Date	Prev Close
2016-01-01	NaN
2016-01-04	NaN
2016-01-05	NaN
2016-01-06	NaN
2016-01-07	NaN
...	...
2018-12-26	13.710389
2018-12-27	14.720972
2018-12-28	14.089607
2018-12-31	14.061916
2019-01-01	13.720232

742 rows × 1 columns

In [26]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock,color="yellow",label="Original")
plt.plot(rollmean,color="red",label="Roll Mean")
plt.plot(rollstd,color="blue",label="Roll Std")
plt.legend()
plt.show()
```



In [28]:

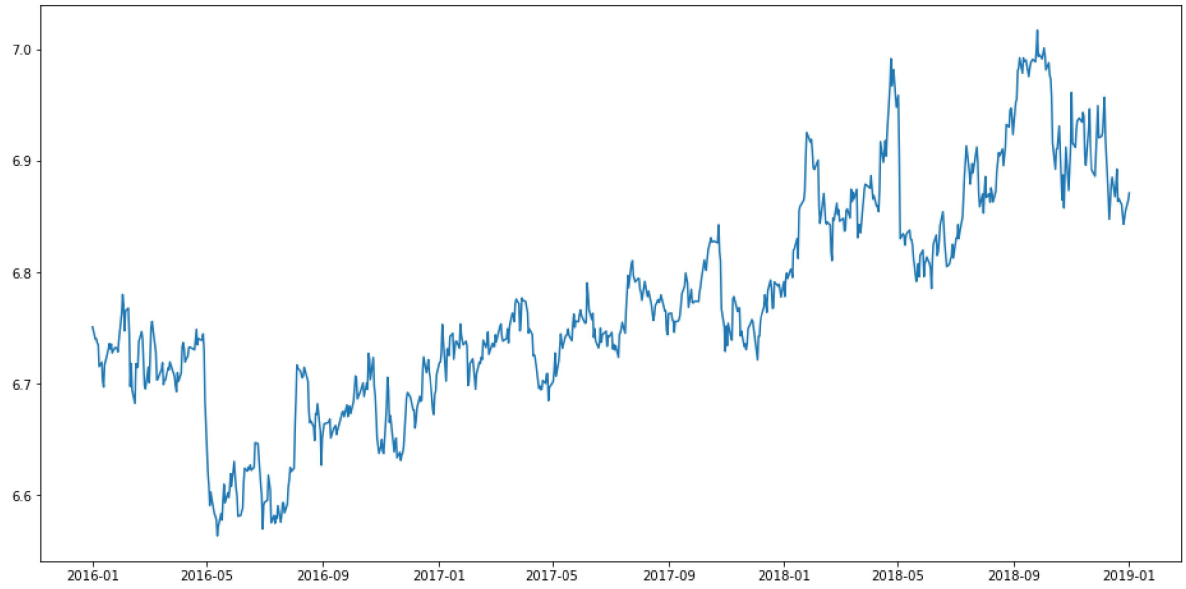
```
dfstock_log=np.log(dfstock)
```

In [34]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock_log)
```

Out[34]:

```
[<matplotlib.lines.Line2D at 0x28409e44280>]
```



In [30]:

```
dfstock_diff=dfstock-dfstock.shift(periods=1)
```

In [31]:

```
dfstock_diff
```

Out[31]:

Date	Prev Close
2016-01-01	NaN
2016-01-04	-9.25
2016-01-05	0.10
2016-01-06	-3.15
2016-01-07	-1.40
...	...
2018-12-26	-16.60
2018-12-27	4.95
2018-12-28	6.00
2018-12-31	9.55
2019-01-01	6.00

742 rows × 1 columns

In [32]:

```
dfstock_diff=dfstock_diff[1:]
```

In [33]:

```
dfstock_diff
```

Out[33]:

Date	Prev Close
2016-01-04	-9.25
2016-01-05	0.10
2016-01-06	-3.15
2016-01-07	-1.40
2016-01-08	-16.35
...	...
2018-12-26	-16.60
2018-12-27	4.95
2018-12-28	6.00
2018-12-31	9.55
2019-01-01	6.00

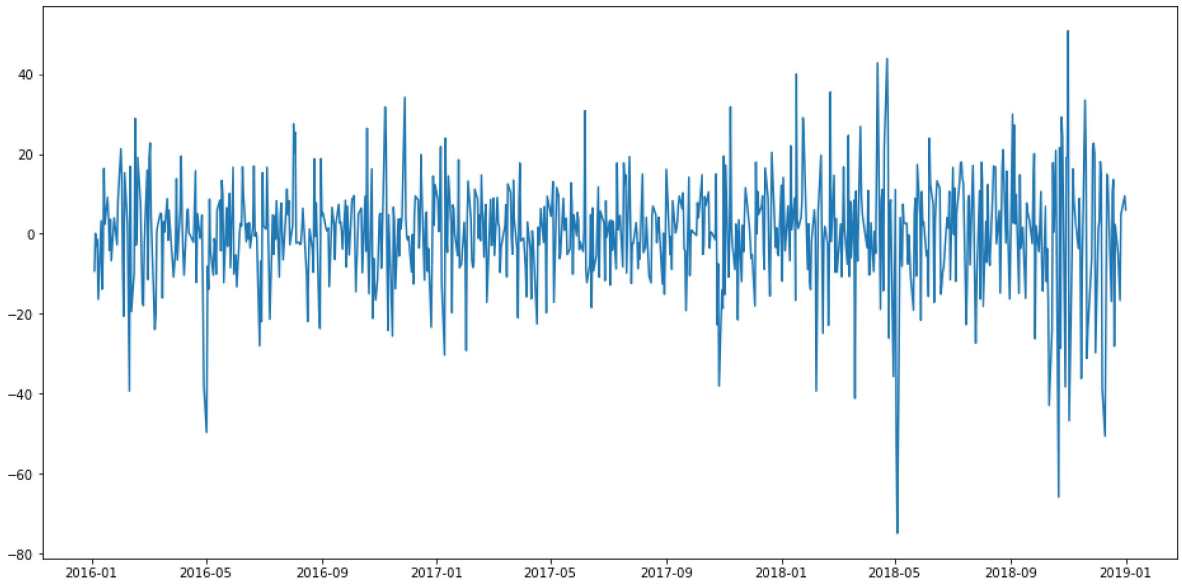
741 rows × 1 columns

In [35]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock_diff)
```

Out[35]:

```
[<matplotlib.lines.Line2D at 0x2840a0d0280>]
```



In [37]:

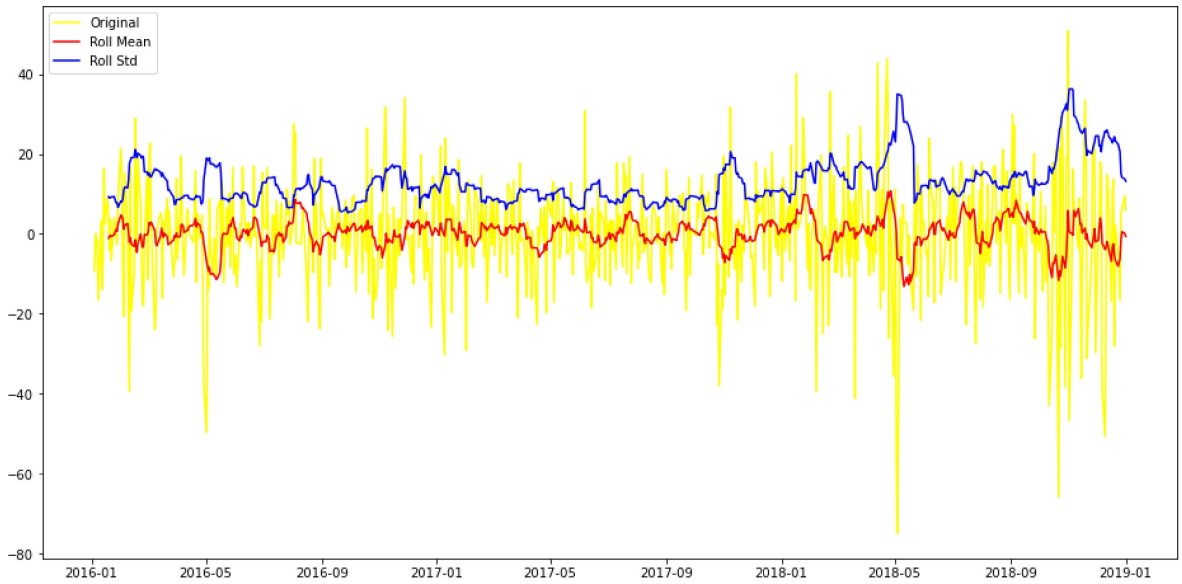
```
dfstock_diff_mean=dfstock_diff.rolling(12).mean()
```

In [38]:

```
dfstock_diff_std=dfstock_diff.rolling(12).std()
```

In [39]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock_diff,color="yellow",label="Original")
plt.plot(dfstock_diff_mean,color="red",label="Roll Mean")
plt.plot(dfstock_diff_std,color="blue",label="Roll Std")
plt.legend()
plt.show()
```



In [40]:

```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
```

In [42]:

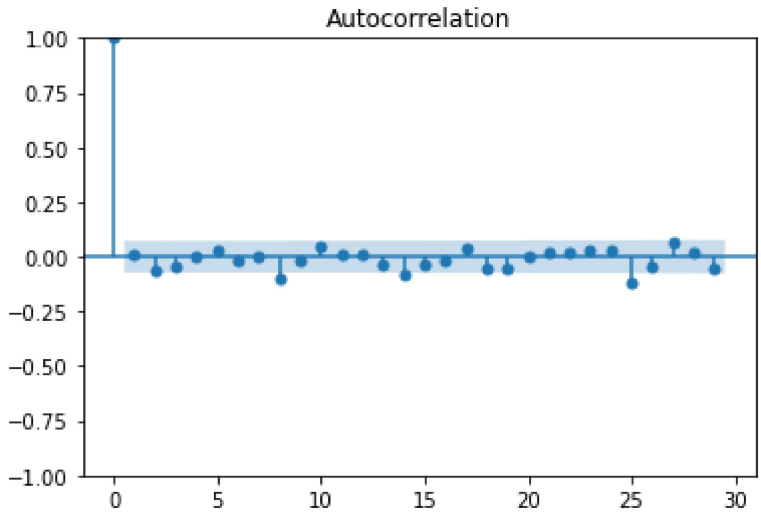
```
import warnings
```

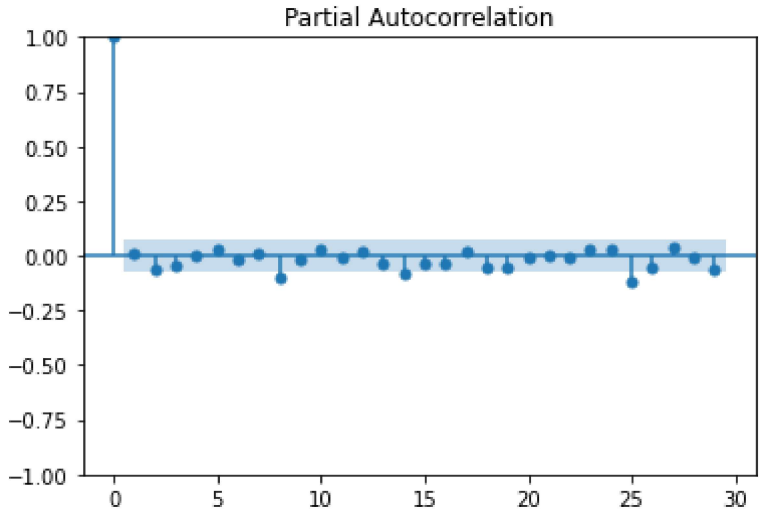
In [43]:

```
warnings.filterwarnings("ignore")
```

In [45]:

```
plot_acf(dfstock_diff)
plot_pacf(dfstock_diff)
plt.show()
```





ARIMA MODEL

In [46]:

```
from statsmodels.tsa.arima.model import ARIMA
```

In [48]:

```
arima_model=ARIMA(dfstock_diff,order=(2,1,2))
```

In [49]:

```
arima_model_fit=arima_model.fit()
```

In [50]:

```
arima_model_fit.forecast(10)
```

Out[50]:

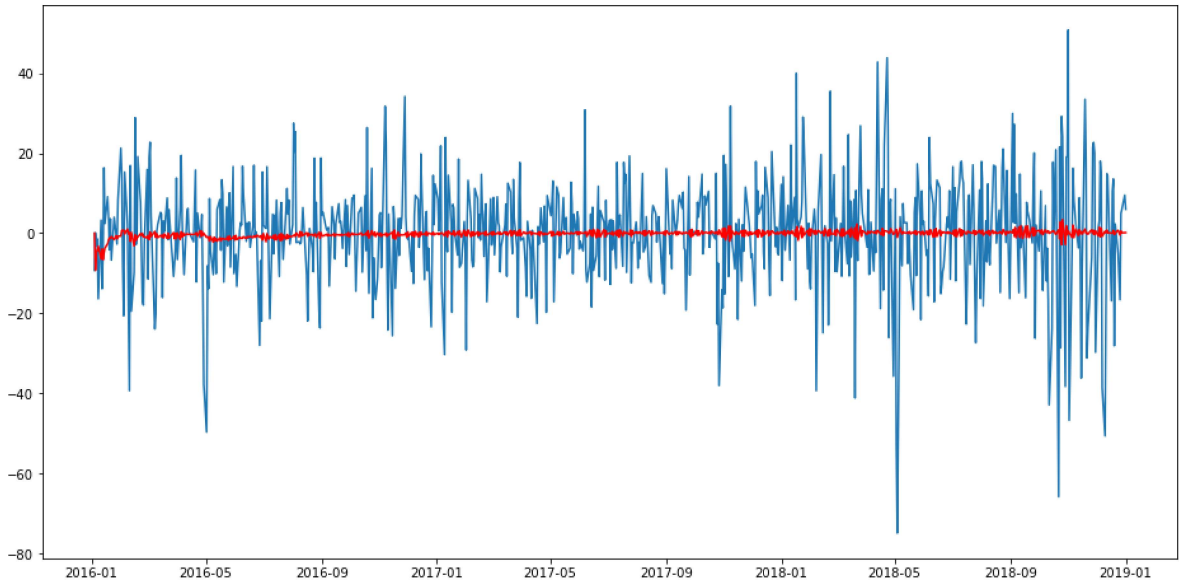
```
741    0.038079
742    0.134145
743    0.159713
744    0.136771
745    0.155381
746    0.140324
747    0.152506
748    0.142651
749    0.150623
750    0.144173
Name: predicted_mean, dtype: float64
```

In [52]:

```
fitvalue=arima_model_fit.fittedvalues
```

In [54]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock_diff)
plt.plot(fitvalue,color="red")
plt.show()
```



Arima Model To Fit Original Data

In [55]:

```
dfstock
```

Out[55]:

Date	Prev Close
2016-01-01	855.10
2016-01-04	845.85
2016-01-05	845.95
2016-01-06	842.80
2016-01-07	841.40
...	...
2018-12-26	937.85
2018-12-27	942.80
2018-12-28	948.80
2018-12-31	958.35
2019-01-01	964.35

742 rows × 1 columns

In [56]:

```
dfstock_diff
```

Out[56]:

Date	Prev Close
2016-01-04	-9.25
2016-01-05	0.10
2016-01-06	-3.15
2016-01-07	-1.40
2016-01-08	-16.35
...	...
2018-12-26	-16.60
2018-12-27	4.95
2018-12-28	6.00
2018-12-31	9.55
2019-01-01	6.00

741 rows × 1 columns

In [57]:

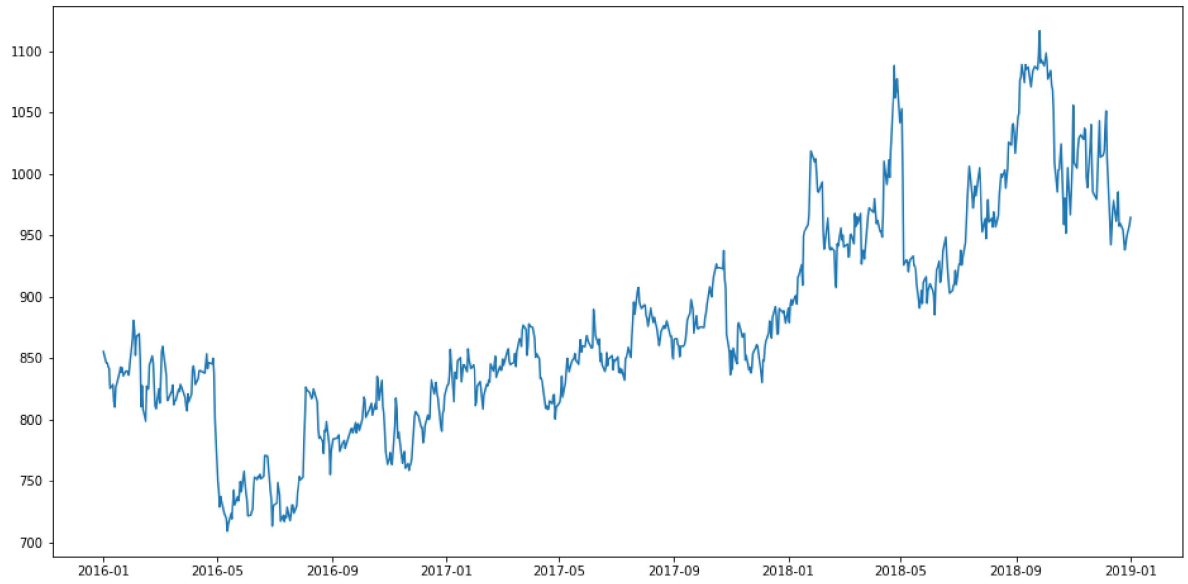
```
from statsmodels.tsa.arima.model import ARIMA
```

In [58]:

```
plt.figure(figsize=(16,8))
plt.plot(dfstock)
```

Out[58]:

```
[<matplotlib.lines.Line2D at 0x284107f6fd0>]
```



In [59]:

```
skyarima=ARIMA(dfstock,order=(2,1,2))
```

In [60]:

```
skyarima_fit=skyarima.fit()
```

In [64]:

```
stock_predict=skyarima_fit.forecast(100)
```

In [65]:

```
stock_predict
```

Out[65]:

```
742    958.728458
743    954.791912
744    957.093784
745    962.958523
746    965.591457
          ...
837    955.212321
838    957.425642
839    962.817873
840    965.141945
841    961.713315
Name: predicted_mean, Length: 100, dtype: float64
```

In [69]:

```
stockdate=pd.date_range(start="2019-01-02",periods=100,freq="B")
```

In [70]:

```
stockdate
```

Out[70]:

```
DatetimeIndex(['2019-01-02', '2019-01-03', '2019-01-04', '2019-01-07',
               '2019-01-08', '2019-01-09', '2019-01-10', '2019-01-11',
               '2019-01-14', '2019-01-15', '2019-01-16', '2019-01-17',
               '2019-01-18', '2019-01-21', '2019-01-22', '2019-01-23',
               '2019-01-24', '2019-01-25', '2019-01-28', '2019-01-29',
               '2019-01-30', '2019-01-31', '2019-02-01', '2019-02-04',
               '2019-02-05', '2019-02-06', '2019-02-07', '2019-02-08',
               '2019-02-11', '2019-02-12', '2019-02-13', '2019-02-14',
               '2019-02-15', '2019-02-18', '2019-02-19', '2019-02-20',
               '2019-02-21', '2019-02-22', '2019-02-25', '2019-02-26',
               '2019-02-27', '2019-02-28', '2019-03-01', '2019-03-04',
               '2019-03-05', '2019-03-06', '2019-03-07', '2019-03-08',
               '2019-03-11', '2019-03-12', '2019-03-13', '2019-03-14',
               '2019-03-15', '2019-03-18', '2019-03-19', '2019-03-20',
               '2019-03-21', '2019-03-22', '2019-03-25', '2019-03-26',
               '2019-03-27', '2019-03-28', '2019-03-29', '2019-04-01',
               '2019-04-02', '2019-04-03', '2019-04-04', '2019-04-05',
               '2019-04-08', '2019-04-09', '2019-04-10', '2019-04-11',
               '2019-04-12', '2019-04-15', '2019-04-16', '2019-04-17',
               '2019-04-18', '2019-04-19', '2019-04-22', '2019-04-23',
               '2019-04-24', '2019-04-25', '2019-04-26', '2019-04-29',
               '2019-04-30', '2019-05-01', '2019-05-02', '2019-05-03',
               '2019-05-06', '2019-05-07', '2019-05-08', '2019-05-09',
               '2019-05-10', '2019-05-13', '2019-05-14', '2019-05-15',
               '2019-05-16', '2019-05-17', '2019-05-20', '2019-05-21'],
              dtype='datetime64[ns]', freq='B')
```

In [67]:

```
dfstock_pred=pd.DataFrame(stock_predict)
```

In [68]:

```
dfstock_pred
```

Out[68]:

	predicted_mean
742	958.728458
743	954.791912
744	957.093784
745	962.958523
746	965.591457
...	...
837	955.212321
838	957.425642
839	962.817873
840	965.141945
841	961.713315

100 rows × 1 columns

In [72]:

```
dfstock_pred["Date"]=stockdate
```

In [73]:

```
dfstock_pred
```

Out[73]:

	predicted_mean	Date
742	958.728458	2019-01-02
743	954.791912	2019-01-03
744	957.093784	2019-01-04
745	962.958523	2019-01-07
746	965.591457	2019-01-08
...	...	...
837	955.212321	2019-05-15
838	957.425642	2019-05-16
839	962.817873	2019-05-17
840	965.141945	2019-05-20
841	961.713315	2019-05-21

100 rows × 2 columns

In [75]:

```
dfstock_pred=dfstock_pred[["Date","predicted_mean"]]
```

In [76]:

```
dfstock_pred
```

Out[76]:

	Date	predicted_mean
742	2019-01-02	958.728458
743	2019-01-03	954.791912
744	2019-01-04	957.093784
745	2019-01-07	962.958523
746	2019-01-08	965.591457
...	...	...
837	2019-05-15	955.212321
838	2019-05-16	957.425642
839	2019-05-17	962.817873
840	2019-05-20	965.141945
841	2019-05-21	961.713315

100 rows × 2 columns

In [ ]:

```

```