In [14]:
```
#importing the python libraries - pandas and numpy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('fivethirtyeight')
```

In [15]:
```
#Importing & reading the dataset
#This creates a dataframe from the CSV file
#Set date as index:

df=pd.read_csv("assignment .csv",index_col='Date',parse_dates=True)
```

In [16]:
```
df
```

Out[16]:

	Ticker	Time	Open	High	Low	Close	SMoving_Average7	SMo
Date								
2020-05-20	NIFTY1	9:16:59	8926.4004	8937.0000	8920.2002	8932.7002	NaN	
2020-05-20	NIFTY1	9:17:59	8933.3496	8944.7998	8933.2998	8937.7500	NaN	
2020-05-20	NIFTY1	9:18:59	8937.4004	8962.0498	8937.4004	8959.7998	NaN	
2020-05-20	NIFTY1	9:19:59	8958.0000	8958.5996	8951.4502	8952.7998	NaN	
2020-05-20	NIFTY1	9:20:59	8955.5498	8979.2998	8953.8496	8974.7998	NaN	
...	...	...	...	...	...	...	...	
2020-10-28	NIFTY1	13:37:59	11743.5996	11743.5996	11736.2998	11739.5000	11743.399971	
2020-10-28	NIFTY1	13:38:59	11741.0000	11741.5996	11735.4004	11737.9004	11742.814314	
2020-10-28	NIFTY1	13:39:59	11738.0000	11740.5996	11736.2998	11739.5996	11742.057057	
2020-10-28	NIFTY1	13:40:59	11741.2002	11741.2002	11733.4004	11736.4004	11740.985629	
2020-10-28	NIFTY1	13:41:59	11735.0996	11735.0996	11715.5000	11720.2998	11737.799943	

42396 rows × 8 columns

In [17]:
```
#Checking if any column has any empty value:
df.isna().sum()
```

Out[17]:

```
Ticker                 0
Time                   0
Open                   0
High                   0
Low                    0
Close                  0
SMoving_Average7       6
SMoving_Average14     13
dtype: int64
```

In [18]:

```
#dropping the empty data entries:
df.dropna(inplace=True)
df.isna().sum()
```

Out[18]:

```
Ticker               0
Time                 0
Open                 0
High                 0
Low                  0
Close                0
SMoving_Average7     0
SMoving_Average14    0
dtype: int64
```

In [19]:

```
df
```

Out[19]:

	Ticker	Time	Open	High	Low	Close	SMoving_Average7	SMov
Date								
2020-05-20	NIFTY1	9:29:59	8963.6504	8963.6504	8950.0000	8952.4502	8961.099886	
2020-05-20	NIFTY1	9:30:59	8953.9502	8953.9502	8945.0000	8950.2500	8958.171314	
2020-05-20	NIFTY1	9:31:59	8950.0498	8961.5000	8946.1504	8960.3496	8957.714143	
2020-05-20	NIFTY1	9:32:59	8961.3496	8967.4502	8952.8496	8956.6504	8958.171314	
2020-05-20	NIFTY1	9:33:59	8957.2002	8964.1504	8954.7500	8964.1504	8958.614257	
...	...	...	...	...	...	...	...	
2020-10-28	NIFTY1	13:37:59	11743.5996	11743.5996	11736.2998	11739.5000	11743.399971	
2020-10-28	NIFTY1	13:38:59	11741.0000	11741.5996	11735.4004	11737.9004	11742.814314	
2020-10-28	NIFTY1	13:39:59	11738.0000	11740.5996	11736.2998	11739.5996	11742.057057	
2020-10-28	NIFTY1	13:40:59	11741.2002	11741.2002	11733.4004	11736.4004	11740.985629	
2020-10-28	NIFTY1	13:41:59	11735.0996	11735.0996	11715.5000	11720.2998	11737.799943	

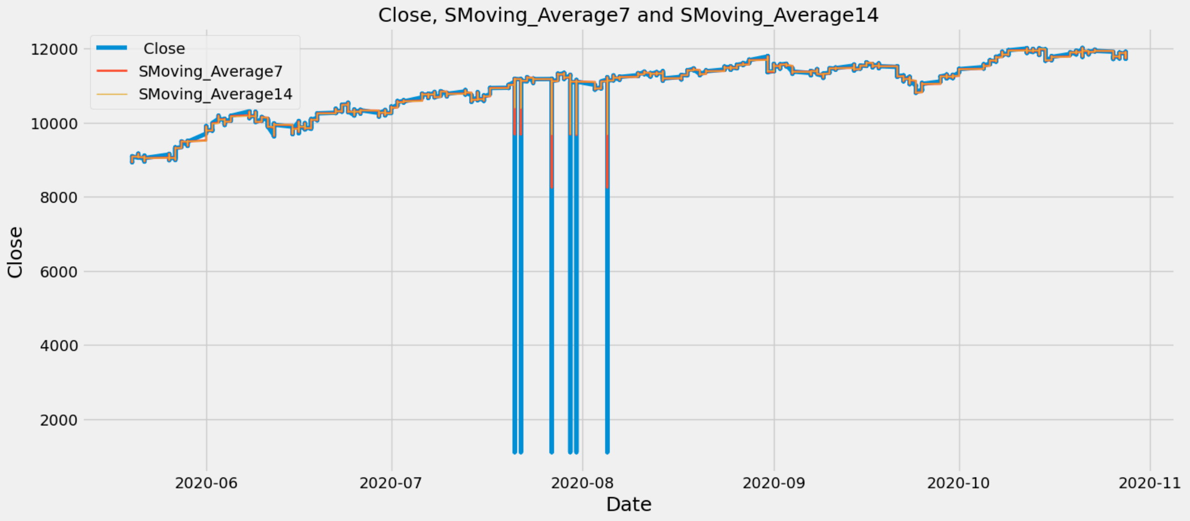
42383 rows × 8 columns

In [23]:

```
#Visualize the SMoving_Average7 and SMoving_Average14
plt.figure(figsize=(16,7))
plt.title('Close, SMoving_Average7 and SMoving_Average14',fontsize=18)
plt.plot(df['Close'],label=' Close',linewidth=4)
plt.plot(df['SMoving_Average7'],label='SMoving_Average7',linewidth=2)
plt.plot(df['SMoving_Average14'],label='SMoving_Average14',linewidth=1)
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close',fontsize=18)
plt.legend()
#plt.show()
```

Out[23]:

```
<matplotlib.legend.Legend at 0x2650676f700>
```



In [27]:

```
def SMA(data, period=30, column='Close'):
    return data[column].rolling(window=period).mean()
```

In [28]:

```
df['SMA30'] = SMA(df)
```

In [29]:

```
df
```

Out[29]:

	Ticker	Time	Open	High	Low	Close	SMoving_Average7	SMo
Date								
2020-05-20	NIFTY1	9:16:59	8926.4004	8937.0000	8920.2002	8932.7002	NaN	
2020-05-20	NIFTY1	9:17:59	8933.3496	8944.7998	8933.2998	8937.7500	NaN	
2020-05-20	NIFTY1	9:18:59	8937.4004	8962.0498	8937.4004	8959.7998	NaN	
2020-05-20	NIFTY1	9:19:59	8958.0000	8958.5996	8951.4502	8952.7998	NaN	
2020-05-20	NIFTY1	9:20:59	8955.5498	8979.2998	8953.8496	8974.7998	NaN	
...	...	...	...	...	...	...	...	
2020-10-28	NIFTY1	13:37:59	11743.5996	11743.5996	11736.2998	11739.5000	11743.399971	
2020-10-28	NIFTY1	13:38:59	11741.0000	11741.5996	11735.4004	11737.9004	11742.814314	
2020-10-28	NIFTY1	13:39:59	11738.0000	11740.5996	11736.2998	11739.5996	11742.057057	
2020-10-28	NIFTY1	13:40:59	11741.2002	11741.2002	11733.4004	11736.4004	11740.985629	
2020-10-28	NIFTY1	13:41:59	11735.0996	11735.0996	11715.5000	11720.2998	11737.799943	

42396 rows × 9 columns

In [24]:
```
df.rename(columns={"SMoving_Average7": "SMA7"}, inplace=True)
```

In [25]:
```
df.rename(columns={"SMoving_Average14": "SMA14"}, inplace=True)
```

In [26]:
```
display()
```

In [27]:
```
df.head(10)
```

Out[27]:

Date	Ticker	Time	Open	High	Low	Close	SMA7	SMA14
2020-05-20	NIFTY1	9:29:59	8963.6504	8963.6504	8950.0000	8952.4502	8961.099886	8960.082029
2020-05-20	NIFTY1	9:30:59	8953.9502	8953.9502	8945.0000	8950.2500	8958.171314	8961.335586
2020-05-20	NIFTY1	9:31:59	8950.0498	8961.5000	8946.1504	8960.3496	8957.714143	8962.949843
2020-05-20	NIFTY1	9:32:59	8961.3496	8967.4502	8952.8496	8956.6504	8958.171314	8962.724886
2020-05-20	NIFTY1	9:33:59	8957.2002	8964.1504	8954.7500	8964.1504	8958.614257	8963.535643
2020-05-20	NIFTY1	9:34:59	8963.5498	8972.3496	8962.9502	8969.5996	8959.792829	8963.164200
2020-05-20	NIFTY1	9:35:59	8968.7998	8970.0000	8960.3496	8964.8496	8959.757114	8961.828471
2020-05-20	NIFTY1	9:36:59	8965.0498	8967.4004	8958.0000	8960.0000	8960.835657	8960.967771
2020-05-20	NIFTY1	9:37:59	8959.2998	8964.9004	8958.5000	8963.4502	8962.721400	8960.446357
2020-05-20	NIFTY1	9:38:59	8963.9004	8973.2998	8963.2002	8973.2998	8964.571429	8961.142786

In [30]:

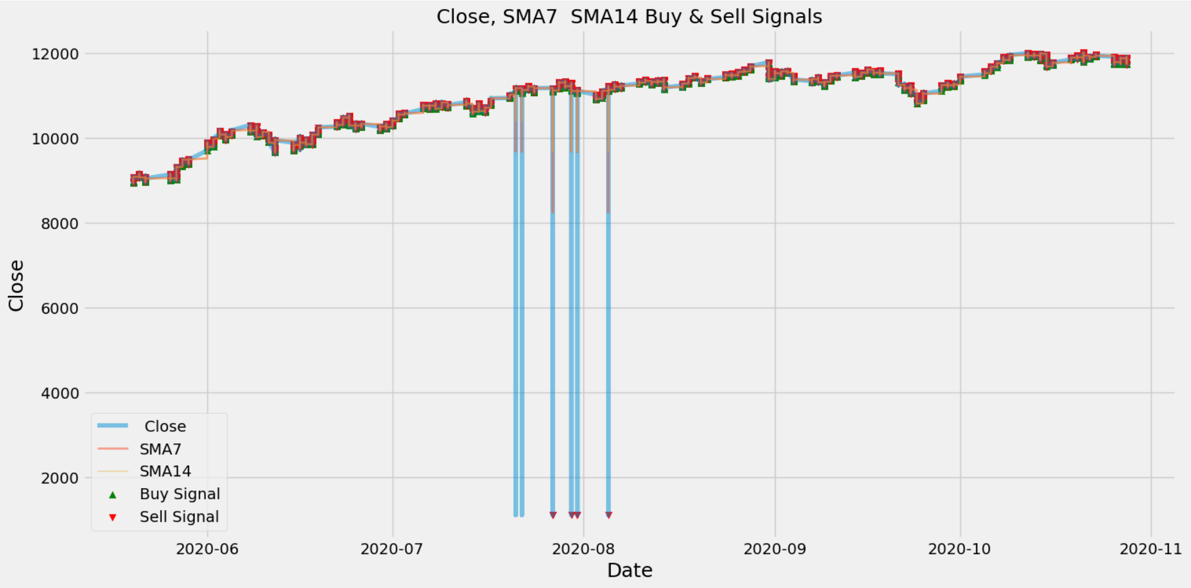
```
#Get the buy and sell signals
df['Signal'] = np.where(df['SMA7']>df['SMA14'],1,0)
df['Position'] = df['Signal'].diff()


df['Buy'] = np.where(df['Position'] == 1 , df['Close'],np.NAN)
df['Sell'] = np.where(df['Position'] == -1 , df['Close'],np.NAN)
```

In [31]:

```
plt.figure(figsize=(16,8))
plt.title('Close, SMA7  SMA14 Buy & Sell Signals',fontsize=18)
plt.plot(df['Close'],alpha= 0.5,label=' Close',linewidth=4)
plt.plot(df['SMA7'],alpha= 0.5,label='SMA7',linewidth=2)
plt.plot(df['SMA14'],alpha= 0.5,label='SMA14',linewidth=1)
plt.scatter(df.index, df['Buy'], alpha = 1, label='Buy Signal',marker = '^',color
plt.scatter(df.index, df['Sell'], alpha = 1, label='Sell Signal',marker = 'v',colo
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close',fontsize=18)
plt.legend()
```

Out[31]:

```
<matplotlib.legend.Legend at 0x26507606490>
```



```
In [32]: df.to_csv("assignment .csv")
```

```
In [33]: df.to_csv("assignment .xlsx")
```

```
In [36]: df.to_csv("assignment .txt",sep="\t")
```

```
In [ ]:
```