```python
[8]: import pandas as pd
     import numpy as np
     import matplotlib as mlb
     import matplotlib.pyplot as plt
     import seaborn as sns
```

```python
[9]: stock_data=pd.read_csv('Stock_Market_Data.csv')
     stock_data.head()
```

[9]:
	Date	Name	Open	High	Low	Close	Volume
0	02-01-2022	01.Bank	22.83	23.20	22.59	22.93	1842350.41
1	03-01-2022	01.Bank	23.03	23.29	22.74	22.90	1664989.63
2	04-01-2022	01.Bank	22.85	23.13	22.64	22.84	1354510.97
3	05-01-2022	01.Bank	22.91	23.20	22.70	22.98	1564334.81
4	06-01-2022	01.Bank	23.12	23.65	23.00	23.37	2586344.19

```python
[10]: stock_data.shape
```

[10]: (49158, 7)

```python
[11]: stock_data.dtypes
```

```
[11]: Date      object
      Name      object
      Open      float64
      High      float64
      Low       float64
      Close     float64
      Volume    float64
      dtype: object
```

```python
[15]: stock_data["Date"]=pd.to_datetime(stock_data["Date"],dayfirst=True)
```

```python
[17]: stock_data["Date"].max()
```

[17]: Timestamp('2022-06-30 00:00:00')

```
[23]: # Creating a list of specific comapnies
      specific_companies=['UNILEVERCL', 'BEXIMCO', 'RECKITTBEN', 'BATBC', 'SAIHAMCOT']
      # Filter out your desired companies' data and put it to another dataframe
      specific_data=stock_data[stock_data['Name'].isin(specific_companies) ]
      # Creating Line Graph for Each Companies
      plt.figure(figsize= (12, 6))
      for company in specific_companies:
          company_data = specific_data[specific_data['Name'] == company]
          plt.plot(company_data['Date'], company_data['Close'], label=company)
      # Adding Labels & Titles
      plt.xlabel('Date')
      plt.ylabel('Closing Value')
      plt.title('Stock Price Trend for Specific Companies')
      plt.legend()
      plt.grid( )
      # Improving readability
      plt.xticks(rotation=45)
      plt.show()
```
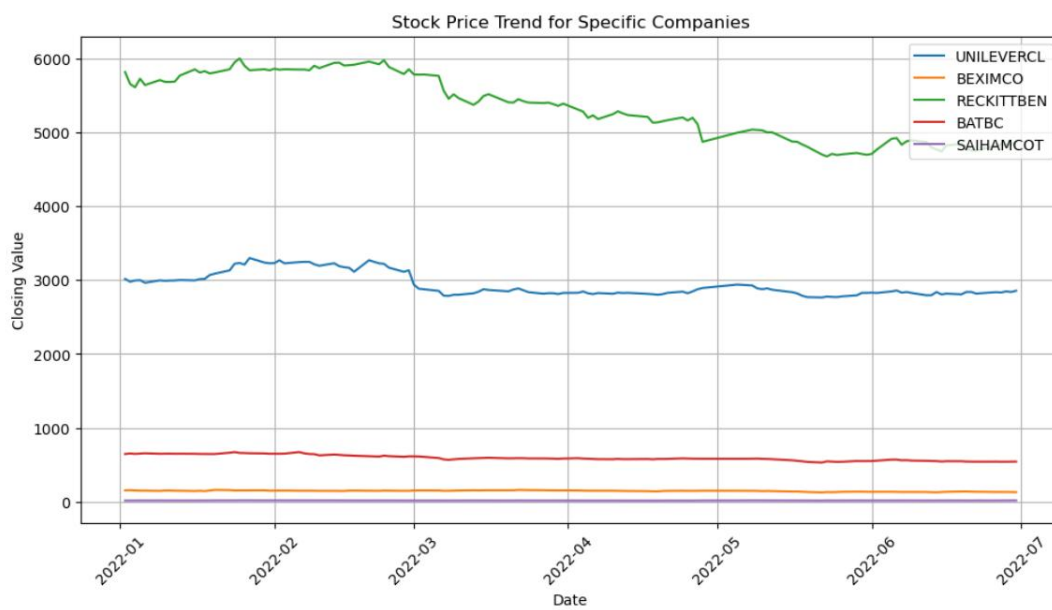
```
[45]: import matplotlib.pyplot as plt
      import pandas as pd

      specific_company = 'RECKITTBEN'
      specific_data = stock_data[stock_data['Name'] == specific_company].copy()  # Use copy() to avoid SettingWithCopyWarning

      # Convert 'Date' to datetime if not done already
      specific_data['Date'] = pd.to_datetime(specific_data['Date'])

      # Calculate 7-day rolling average
      specific_data['7_Day_Rolling_Avg'] = specific_data['Close'].rolling(window=7).mean()

      plt.figure(figsize=(12, 6))
      plt.plot(specific_data['Date'], specific_data['Close'], label=f'{specific_company} Closing Price', color='blue')
      plt.plot(specific_data['Date'], specific_data['7_Day_Rolling_Avg'], label=f'{specific_company} 7 Day Rolling Average of closing price', color='red')

      plt.xlabel('Date')
      plt.ylabel('Closing Price')
      plt.title(f'{specific_company} 7 Day Rolling Average of Closing Price')
      plt.grid()
      plt.legend()
      plt.xticks(rotation=45)
      plt.show()
```


RECKITTBEN 7 Day Rolling Average of Closing Price

```
plt.figure(figsize=(12, 6))
```