DATE - 31/10/2023

PHASE - III

TEAM ID - 719

PROJECT TITLE - AIR QUALITY ANALYSIS IN TAMIL NADU

IMPORTING MODULES

In [4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import tkinter as tk
import random
import requests
import scipy
import xgboost
```

In [5]:
```
dataset = pd.read_csv("datafile.csv")
```

In [6]:
```
import os
print("Current working directory:", os.getcwd())

file_path = 'datafile.csv'
if os.path.exists(file_path):
    print("The file exists.")
else:
    print("The file does not exist at the specified path.")
```

```
Current working directory: C:\Users\VIJAYRAJ R
The file exists.
```

IMPORT THE DATA SET

In [6]:
```
dataset
```

Out[6]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.(
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.(
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.(
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.(
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.(
...	...	...	...	...	...	...	...	...	..
2874	773	12-03-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.(
2875	773	12-10-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.(
2876	773	17-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.(
2877	773	24-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.(
2878	773	31-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.(

2879 rows × 11 columns

In [15]:
```
dataset.head()
```

Out[15]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RS
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	

In [16]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Stn Code                        2879 non-null   int64  
 1   Sampling Date                   2879 non-null   object 
 2   State                           2879 non-null   object 
 3   City/Town/Village/Area          2879 non-null   object 
 4   Location of Monitoring Station  2879 non-null   object 
 5   Agency                          2879 non-null   object 
 6   Type of Location                2879 non-null   object 
 7   SO2                             2868 non-null   float64
 8   NO2                             2866 non-null   float64
 9   RSPM/PM10                       2875 non-null   float64
 10  PM 2.5                          0 non-null      float64
dtypes: float64(4), int64(1), object(6)
memory usage: 247.5+ KB
```

In [17]:
```
dataset.describe()
```

Out[17]:

	Stn Code	SO2	NO2	RSPM/PM10	PM 2.5
count	2879.000000	2868.000000	2866.000000	2875.000000	0.0
mean	475.750261	11.503138	22.136776	62.494261	NaN
std	277.675577	5.051702	7.128694	31.368745	NaN
min	38.000000	2.000000	5.000000	12.000000	NaN
25%	238.000000	8.000000	17.000000	41.000000	NaN
50%	366.000000	12.000000	22.000000	55.000000	NaN
75%	764.000000	15.000000	25.000000	78.000000	NaN
max	773.000000	49.000000	71.000000	269.000000	NaN

In [58]:
```
print(data.isna())
```

```
      Stn Code  Sampling Date  State  City/Town/Village/Area  \
0        False          False  False                   False
1        False          False  False                   False
2        False          False  False                   False
3        False          False  False                   False
4        False          False  False                   False
...        ...            ...    ...                     ...
2874     False          False  False                   False
2875     False          False  False                   False
2876     False          False  False                   False
2877     False          False  False                   False
2878     False          False  False                   False

      Location of Monitoring Station  Agency  Type of Location    SO2    NO2  \
0                              False   False             False  False  False
1                              False   False             False  False  False
2                              False   False             False  False  False
3                              False   False             False  False  False
4                              False   False             False  False  False
...                              ...     ...               ...    ...    ...
2874                           False   False             False  False  False
2875                           False   False             False  False  False
2876                           False   False             False  False  False
2877                           False   False             False  False  False
2878                           False   False             False  False  False

      RSPM/PM10  PM 2.5
0         False    True
1         False    True
2         False    True
3         False    True
4         False    True
...         ...     ...
2874      False    True
2875      False    True
2876      False    True
2877      False    True
2878      False    True

[2879 rows x 11 columns]
```

In [59]:
```
print(data.isna().any())
```

```
Stn Code                          False
Sampling Date                     False
State                             False
City/Town/Village/Area            False
Location of Monitoring Station    False
Agency                            False
Type of Location                  False
SO2                                True
NO2                                True
RSPM/PM10                          True
PM 2.5                             True
dtype: bool
```

In [25]:
```
import pandas as pd
dataset = pd.read_csv('datafile.csv')
numeric_dataset = dataset.select_dtypes(include=[np.number])
correlation_matrix = numeric_dataset.corr()

print(correlation_matrix)
```

```
           Stn Code       SO2       NO2  RSPM/PM10  PM 2.5
Stn Code   1.000000  0.263537 -0.043257   0.336190     NaN
SO2        0.263537  1.000000  0.078246   0.445152     NaN
NO2       -0.043257  0.078246  1.000000   0.068277     NaN
RSPM/PM10  0.336190  0.445152  0.068277   1.000000     NaN
PM 2.5          NaN       NaN       NaN        NaN     NaN
```

In [35]:
```
import pandas as pd
data = pd.read_csv('datafile.csv')
data = data.drop(columns=['PM 2.5'])
```

In [36]:
```
data
```

Out[36]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.(
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.(
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.(
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.(
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.(
...	...	...	...	...	...	...	...	...	..
2874	773	12-03-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.(
2875	773	12-10-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.(
2876	773	17-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.(
2877	773	24-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.(
2878	773	31-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.(

2879 rows × 10 columns

In [38]:
```
mean_SO2 = data['SO2'].mean()
mean_NO2 = data['NO2'].mean()
mean_RSPM_PM10 = data['RSPM/PM10'].mean()
```

In [39]:
```
mean_SO2,mean_NO2, mean_RSPM_PM10
```

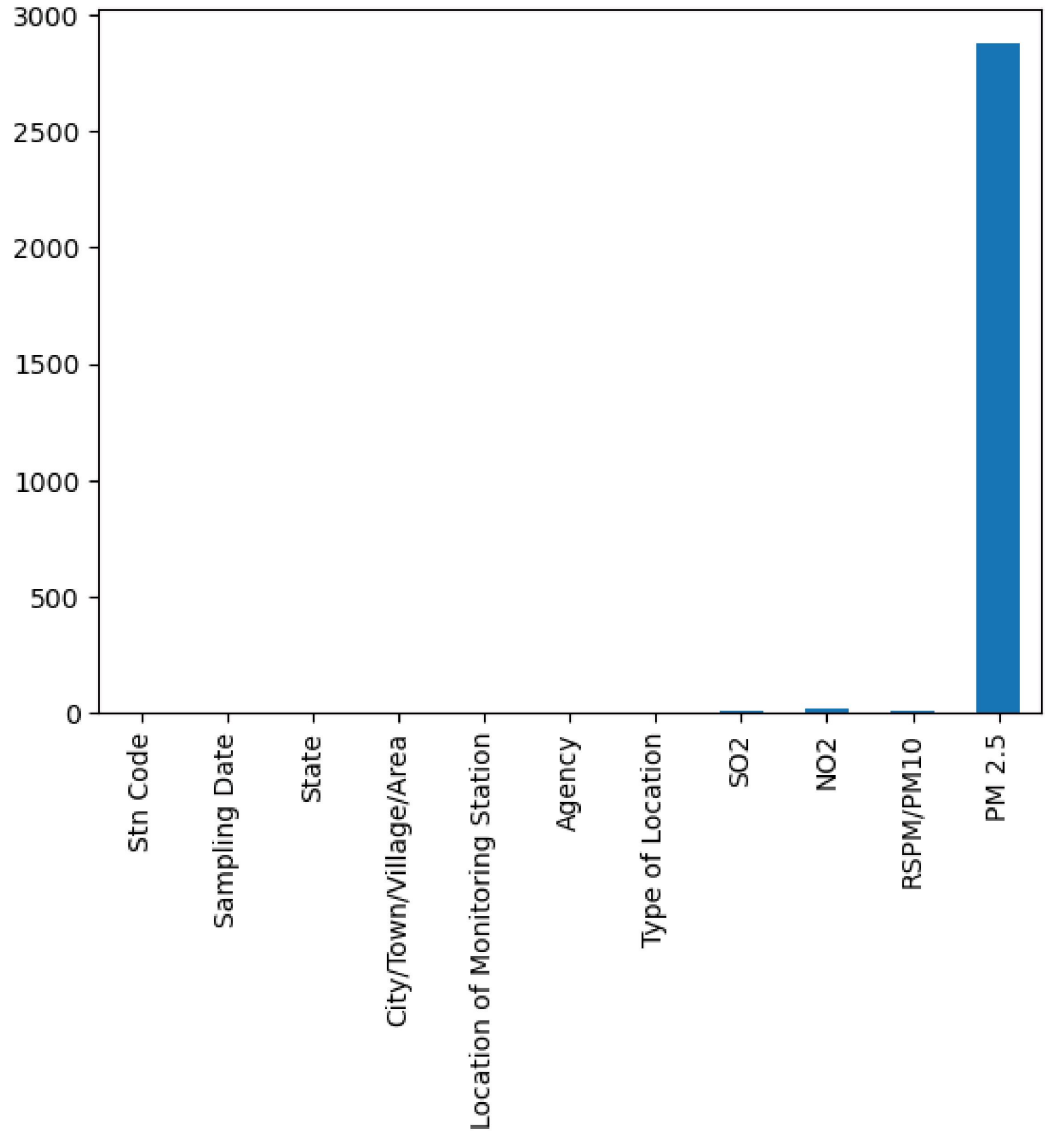
Out[39]:
```
(11.503138075313808, 22.136775994417306, 62.494260869565224)
```

In [40]:
```
data['SO2'].fillna(value=mean_SO2,inplace=True)
data['NO2'].fillna(value=mean_NO2,inplace=True)
data['RSPM/PM10'].fillna(value=mean_RSPM_PM10,inplace=True)
```

PREPROCSSING DATA

In [33]:
```
dataset.isna().sum().plot(kind='bar')
```

Out[33]:
```
<Axes: >
```

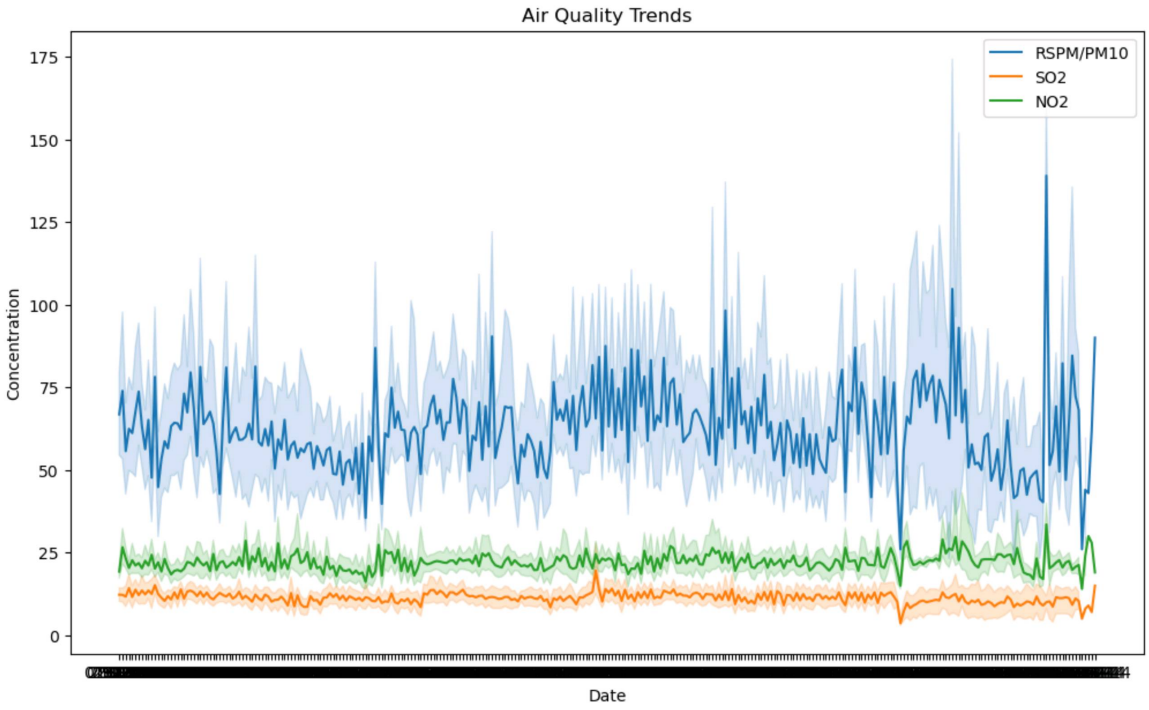


In [41]:
```
cleandata=new_data.isnull().sum()
```

In [42]:
```
cleandata
```

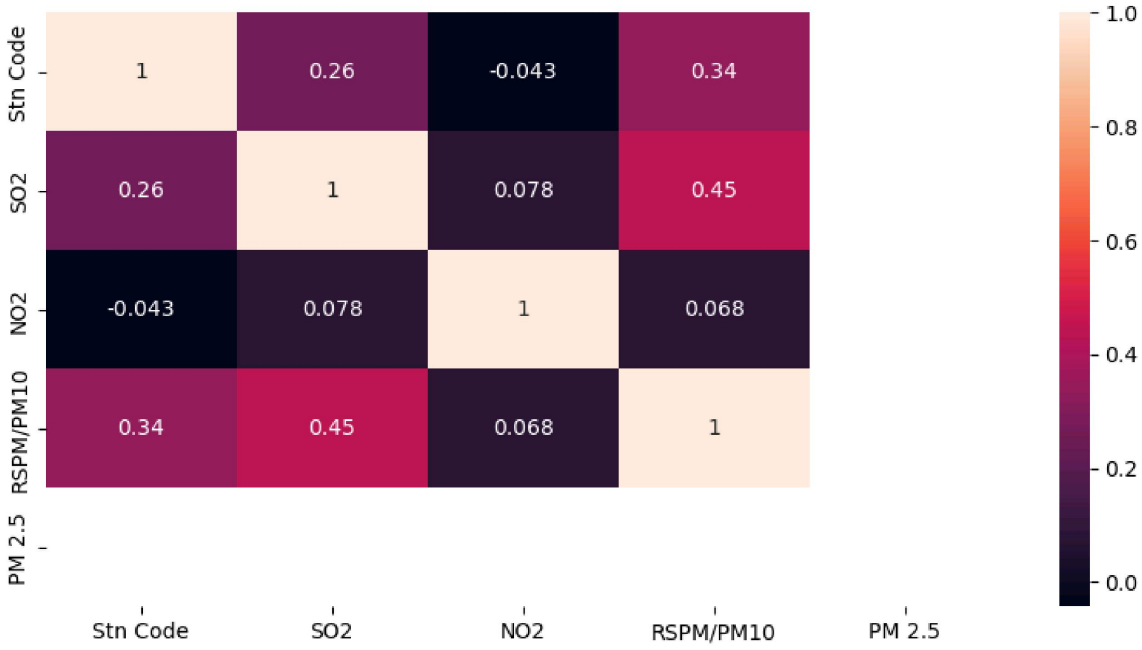
Out[42]:
```
Stn Code                          0
Sampling Date                     0
State                             0
City/Town/Village/Area            0
Location of Monitoring Station    0
Agency                            0
Type of Location                  0
SO2                               0
NO2                               0
RSPM/PM10                         0
dtype: int64
```

In [43]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12, 7))
sns.lineplot(data=new_data, x='Sampling Date', y='RSPM/PM10', label='RSPM/PM10')
sns.lineplot(data=new_data, x='Sampling Date', y='SO2', label='SO2')
sns.lineplot(data=new_data, x='Sampling Date', y='NO2', label='NO2')
plt.title('Air Quality Trends')
plt.xlabel('Date')
plt.ylabel('Concentration')
plt.legend()
plt.show()
```



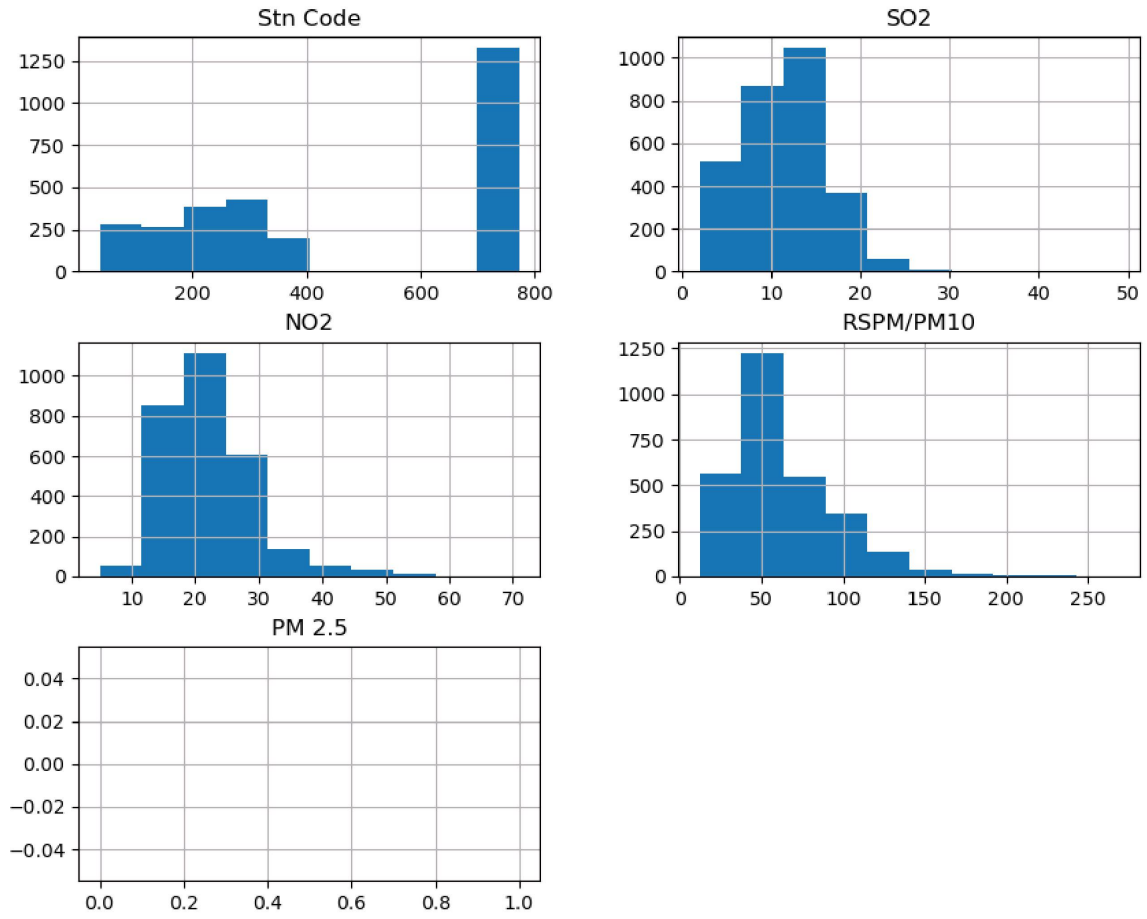
In [28]:
```
plt.figure(figsize=(10,5))
sns.heatmap(dataset.corr(numeric_only = True), annot=True)
```

Out[28]: <Axes: >



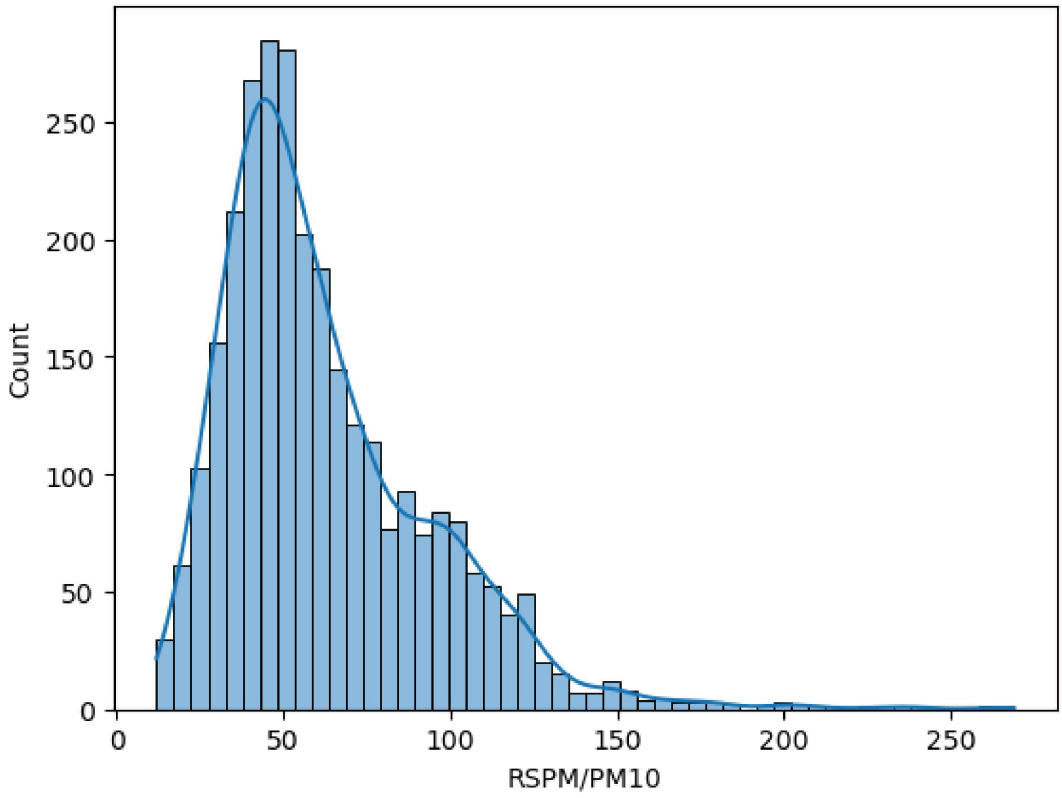
In [29]:
```
dataset.hist(figsize=(10,8))
```

Out[29]:
```
array([[<Axes: title={'center': 'Stn Code'}>,
        <Axes: title={'center': 'SO2'}>],
       [<Axes: title={'center': 'NO2'}>,
        <Axes: title={'center': 'RSPM/PM10'}>],
       [<Axes: title={'center': 'PM 2.5'}>, <Axes: >]], dtype=object)
```



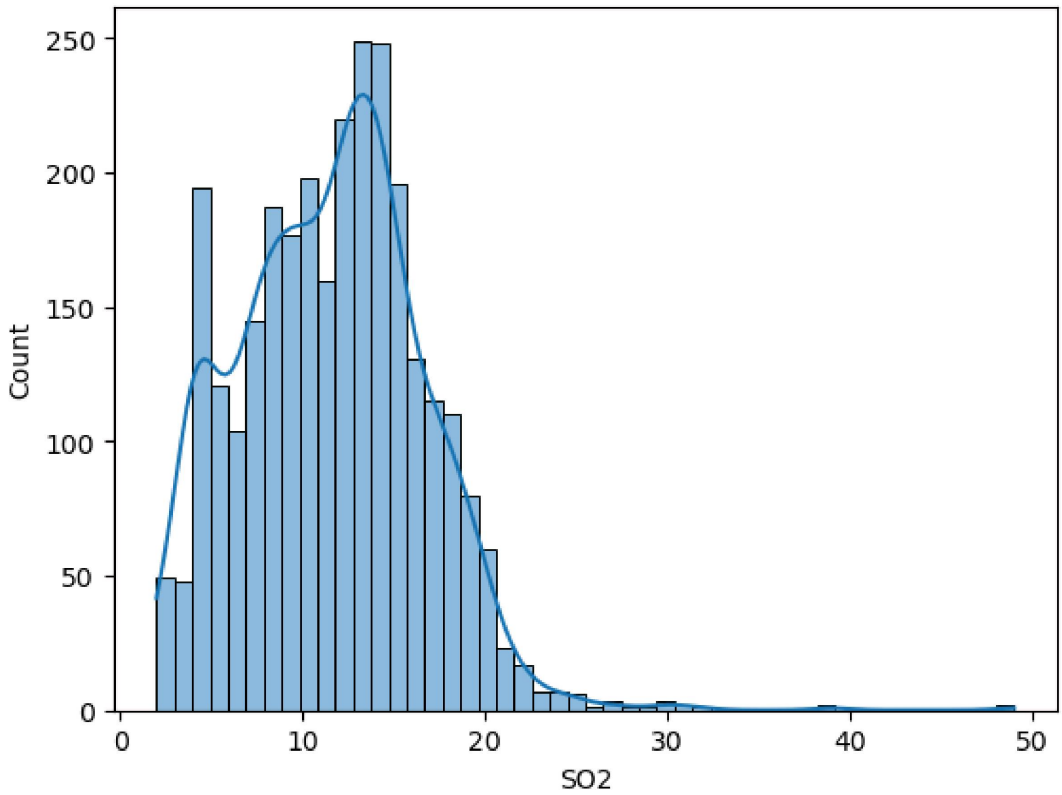
In [30]:
```
import seaborn as sns
sns.histplot(data=dataset, x="RSPM/PM10", kde=True)
```

Out[30]: <Axes: xlabel='RSPM/PM10', ylabel='Count'>



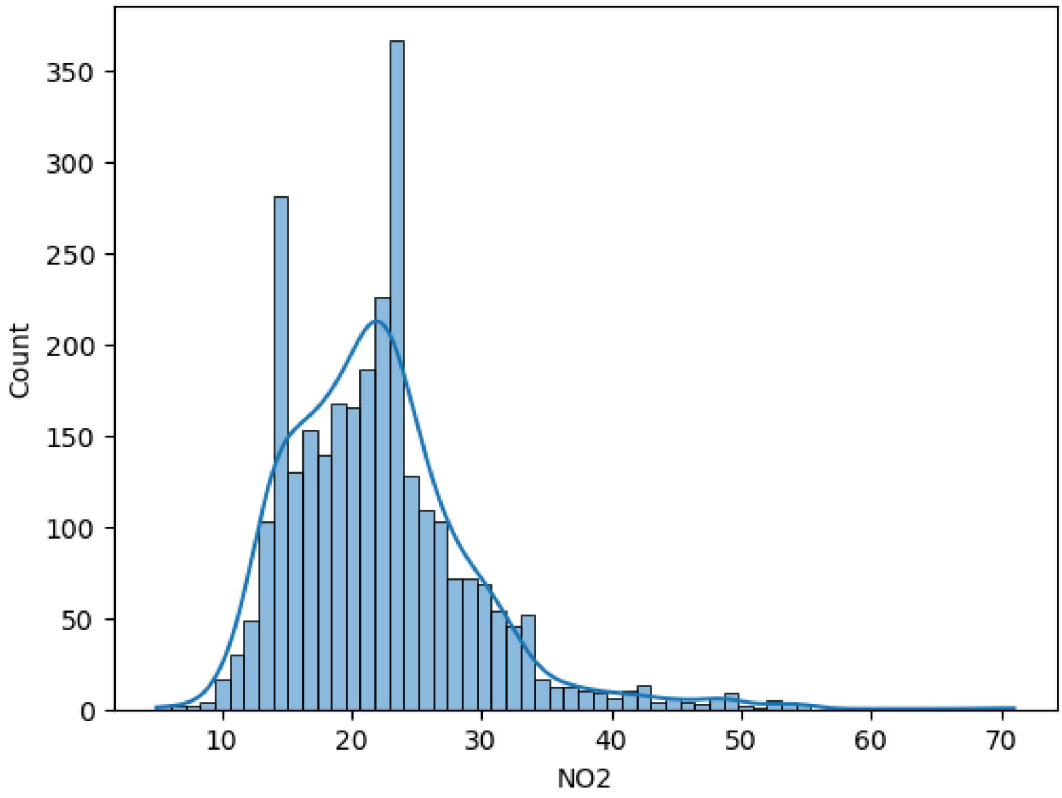
In [32]:
```
import seaborn as sns
sns.histplot(data=dataset, x="SO2", kde=True)
```

Out[32]: <Axes: xlabel='SO2', ylabel='Count'>



In [34]:
```
import seaborn as sns
sns.histplot(data=dataset, x="NO2", kde=True)
```

Out[34]: <Axes: xlabel='NO2', ylabel='Count'>



In [ ]:
```
```

In [13]:
```
import pandas as pd

data = pd.read_csv('datafile.csv')

grouped = data.groupby('City/Town/Village/Area')[['SO2', 'NO2', 'RSPM/PM10']].me

# Display the calculated averages
print(grouped)
```

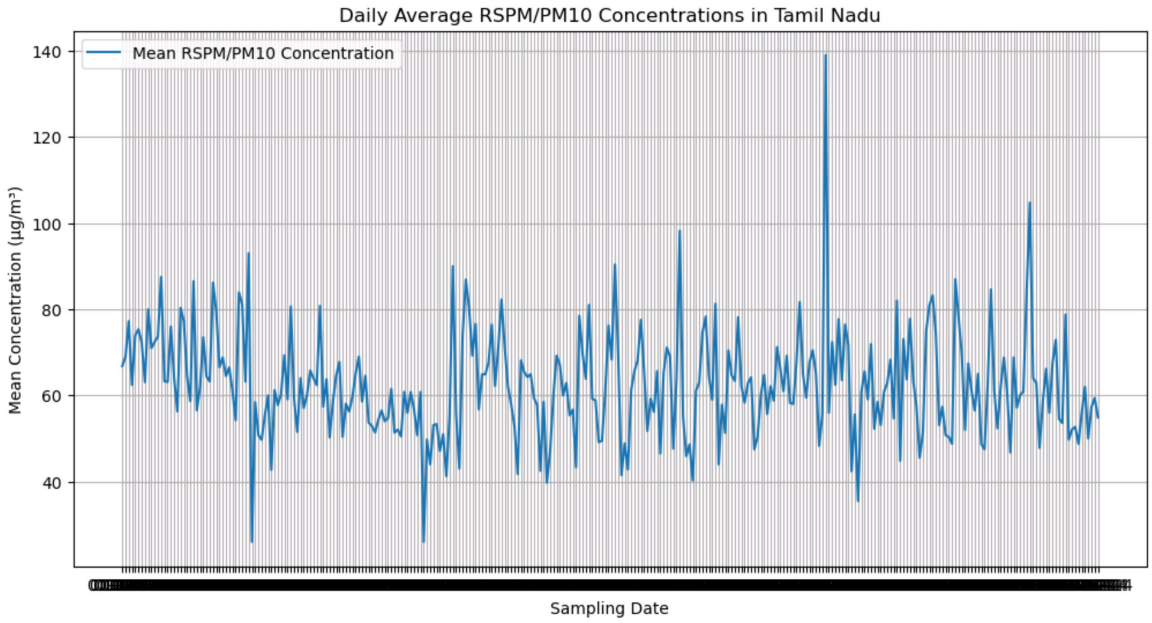
City/Town/Village/Area	SO2	NO2	RSPM/PM10
Chennai	13.014042	22.088442	58.998000
Coimbatore	4.541096	25.325342	49.217241
Cuddalore	8.965986	19.710884	61.881757
Madurai	13.319728	25.768707	45.724490
Mettur	8.429268	23.185366	52.721951
Salem	8.114504	28.664122	62.954198
Thoothukudi	12.989691	18.512027	83.458904
Trichy	15.293956	18.695055	85.054496

In [14]:
```
import pandas as pd

# Load your CSV dataset into a DataFrame
data = pd.read_csv('datafile.csv')

# Group the data by the 'Region' column and calculate the mean for each group
grouped = data.groupby('Location of Monitoring Station')[['SO2', 'NO2', 'RSPM/PM

# Display the calculated averages
print(grouped)
```

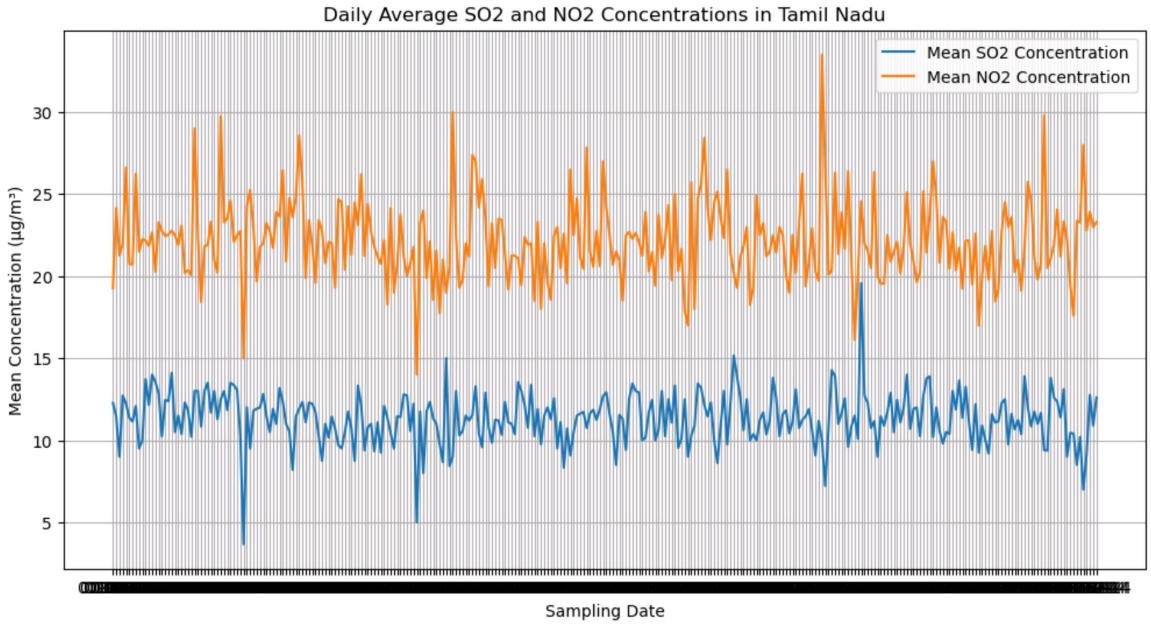
Location of Monitoring Station	SO2	NO2	RSPM/PM10
AVM Jewellery Building, Tuticorin	9.302083	12.697917	70.175258
Adyar, Chennai	13.252174	18.965217	57.068966
Anna Nagar, Chennai	13.873874	20.754545	72.187500
Bishop Heber College, Tirchy	11.800000	14.942857	45.633803
Central Bus Stand, Trichy	18.013333	21.506667	120.546667
District Environmental Engineer Office, Imperia...	8.101010	19.151515	64.020202
Distt. Collector's Office, Coimbatore	4.554348	25.793478	42.322222
Eachangadu Villagae	11.916667	22.395833	75.591837
Fenner (I) Ltd. Employees Assiciation Building ...	13.643564	27.198020	40.732673
Fisheries College, Tuticorin	14.526882	20.204301	85.255319
Gandhi Market, Trichy	17.148649	20.797297	101.743243
Golden Rock, Trichy	12.014085	15.000000	46.222222
Govt. High School, Manali, Chennai.	13.043011	15.408602	44.612903
Highway (Project -I) Building, Madurai	11.947917	24.458333	46.427083
Kathivakkam, Municipal Kalyana Mandapam, Chennai	12.925532	15.170213	46.851064
Kilpauk, Chennai	19.232759	27.172414	88.103448
Kunnathur Chatram East Avani Mollai Street, Mad...	14.340206	25.577320	50.226804
Madras Medical College, Chennai	7.418605	27.465116	35.837209
Main Guard Gate, Tirchy	17.135135	20.837838	107.693333
NEERI, CSIR Campus Chennai	5.931034	23.758621	43.678161
Poniarajapuram, On the top of DEL, Coimbatore	4.126214	23.019417	48.883495
Raja Agencies, Tuticorin	15.058824	22.441176	94.544554
Raman Nagar, Mettur	7.572816	20.407767	51.106796
SIDCO Industrial Complex, Mettur	9.294118	25.990196	54.352941
SIDCO Office, Coimbatore	4.969072	27.329897	55.969072
SIPCOT Industrial Complex, Cuddalore	6.969697	17.666667	46.171717
Sowdeswari College Building, Salem	8.114504	28.664122	62.954198
Thiruvottiyur Municipal Office, Chennai	8.360465	28.069767	34.310345
Thiruvottiyur, Chennai	13.010417	15.583333	
Thiyagaraya Nagar, Chennai	18.849558	28.250000	

```
Thiruvottiyur,  Chennai                    42.604167
Thiyagaraya Nagar, Chennai                102.327434
```

In [15]:
```
# Calculate daily average RSPM/PM10 concentrations for all monitoring stations
daily_mean = data.groupby('Sampling Date')[['RSPM/PM10']].mean()

# Plot daily average RSPM/PM10 concentrations
plt.figure(figsize=(12, 6))
plt.plot(daily_mean.index, daily_mean['RSPM/PM10'], label='Mean RSPM/PM10 Concen
plt.xlabel('Sampling Date')
plt.ylabel('Mean Concentration (µg/m³)')  # Units may vary based on your data
plt.title('Daily Average RSPM/PM10 Concentrations in Tamil Nadu')
plt.legend()
plt.grid(True)
plt.show()
```



In [16]:
```
# Calculate daily average SO2 and NO2 concentrations for all monitoring stations
daily_mean = data.groupby('Sampling Date')[['SO2', 'NO2']].mean()

# Plot daily average SO2 and NO2 concentrations
plt.figure(figsize=(12, 6))
plt.plot(daily_mean.index, daily_mean['SO2'], label='Mean SO2 Concentration')
plt.plot(daily_mean.index, daily_mean['NO2'], label='Mean NO2 Concentration')
plt.xlabel('Sampling Date')
plt.ylabel('Mean Concentration (µg/m³)')  # Units may vary based on your data
plt.title('Daily Average SO2 and NO2 Concentrations in Tamil Nadu')
plt.legend()
plt.grid(True)
plt.show()
```

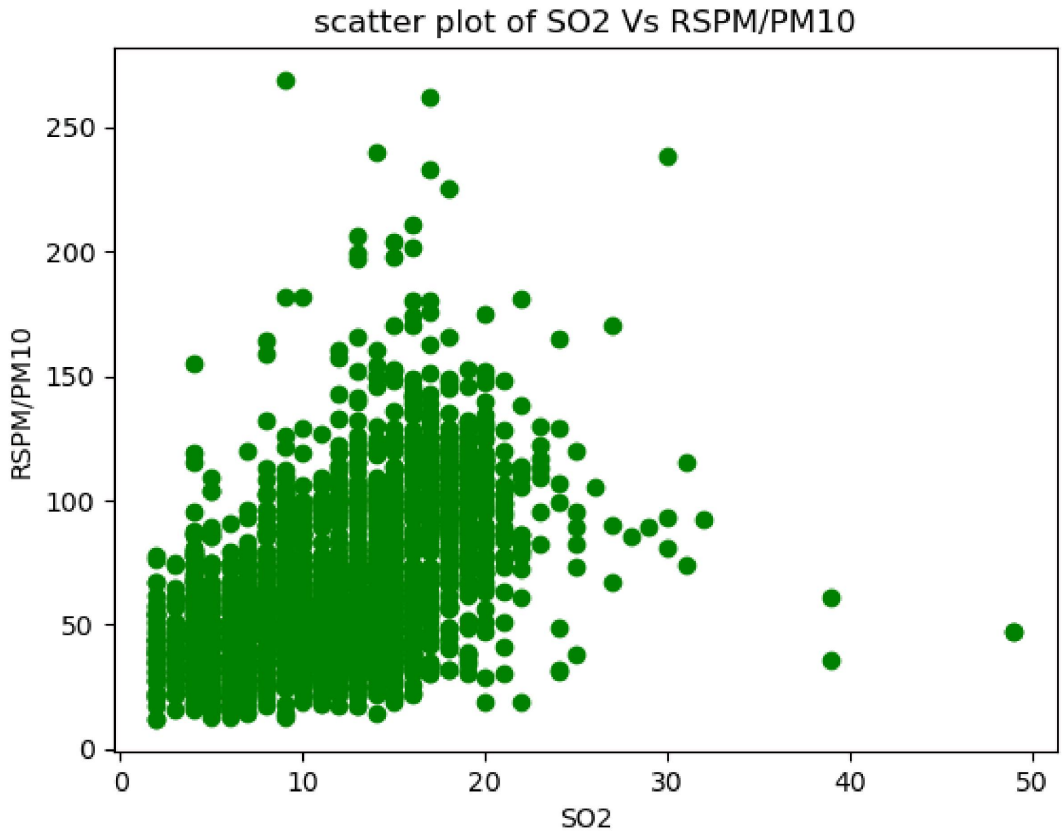


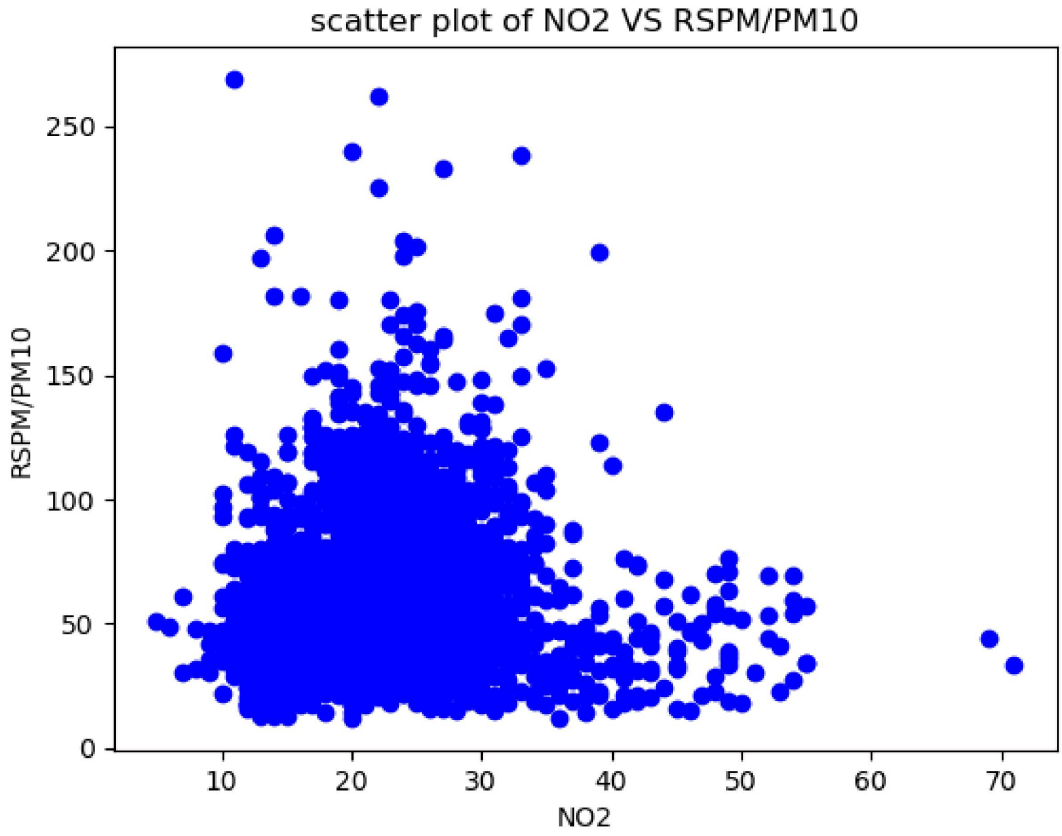
In [27]:
```
print(merged_data.columns)
```

```
Index(['Stn Code_x', 'Sampling Date_x', 'State_x', 'City/Town/Village/Area_x',
       'Location of Monitoring Station_x', 'Agency_x', 'Type of Location_x',
       'SO2_x', 'NO2_x', 'RSPM/PM10_x', 'PM 2.5_x', 'geometry', 'Stn Code_y',
       'Sampling Date_y', 'State_y', 'City/Town/Village/Area_y',
       'Location of Monitoring Station_y', 'Agency_y', 'Type of Location_y',
       'SO2_y', 'NO2_y', 'RSPM/PM10_y', 'PM 2.5_y'],
      dtype='object')
```

In [43]:
```
mean_SO2 = dataset['SO2'].mean()
mean_NO2 = dataset['NO2'].mean()
mean_RSPM_PM10 = dataset['RSPM/PM10'].mean()

average = dataset.groupby(['Location of Monitoring Station', 'City/Town/Village/

average_mean = average.mean()
```

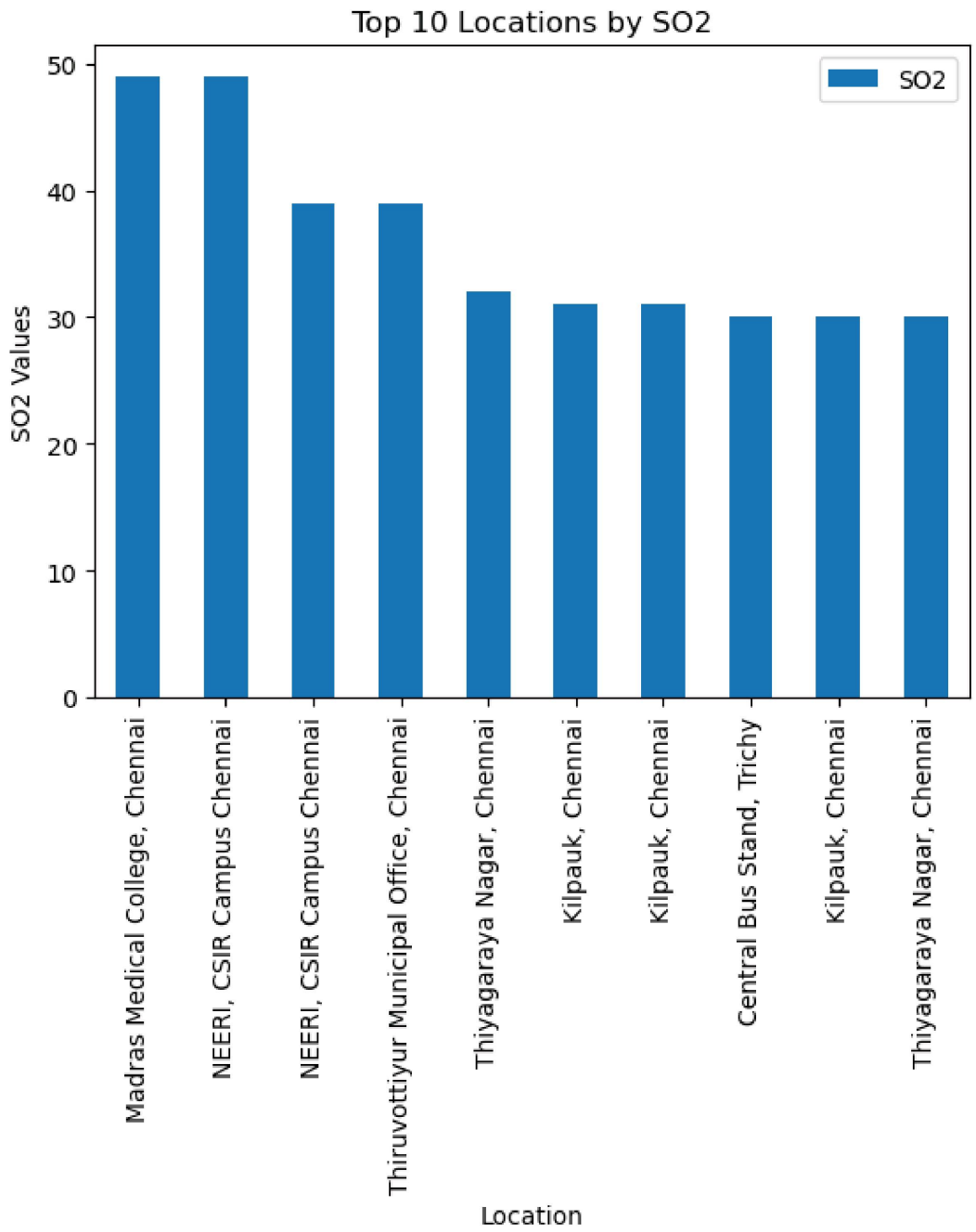
In [48]:
```
fig=plt.figure()

plt.scatter (dataset['SO2'], dataset['RSPM/PM10'], color = 'green')

plt.xlabel("SO2")

plt.ylabel("RSPM/PM10")

plt.title("scatter plot of SO2 Vs RSPM/PM10")

plt.grid(False)

plt.show()

fig=plt.figure()

plt.scatter (dataset['NO2'], dataset['RSPM/PM10'], color = 'blue')

plt.xlabel("NO2")
plt.ylabel("RSPM/PM10")
plt.title("scatter plot of NO2 VS RSPM/PM10")
plt.grid(False)
plt.show()
```




scatter plot of NO2 VS RSPM/PM10
RSPM/PM10
NO2
0
50
100
150
200
250
10
20
30
40
50
60
70

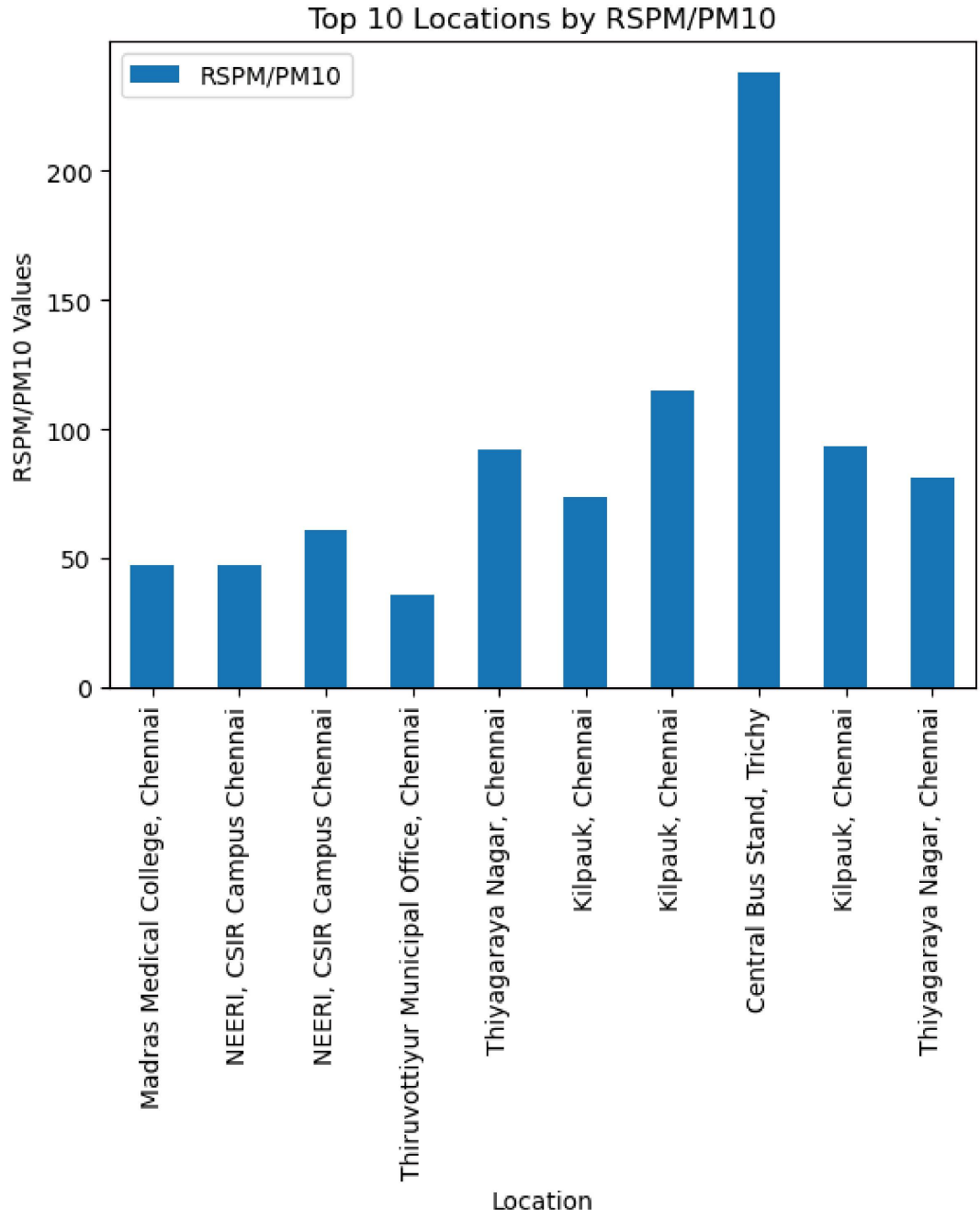
In [56]:
```
import matplotlib.pyplot as plt

# Assuming 'new_data' is your DataFrame and 'SO2' is the column representing sul
plt.figure(figsize=(8, 6))

max_SO2 = dataset.sort_values(by='SO2', ascending=False)
max_SO2.head(10).plot(x='Location of Monitoring Station', y='SO2', kind='bar')

plt.title('Top 10 Locations by SO2')
plt.xlabel('Location')
plt.ylabel('SO2 Values')
plt.grid(False)
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```

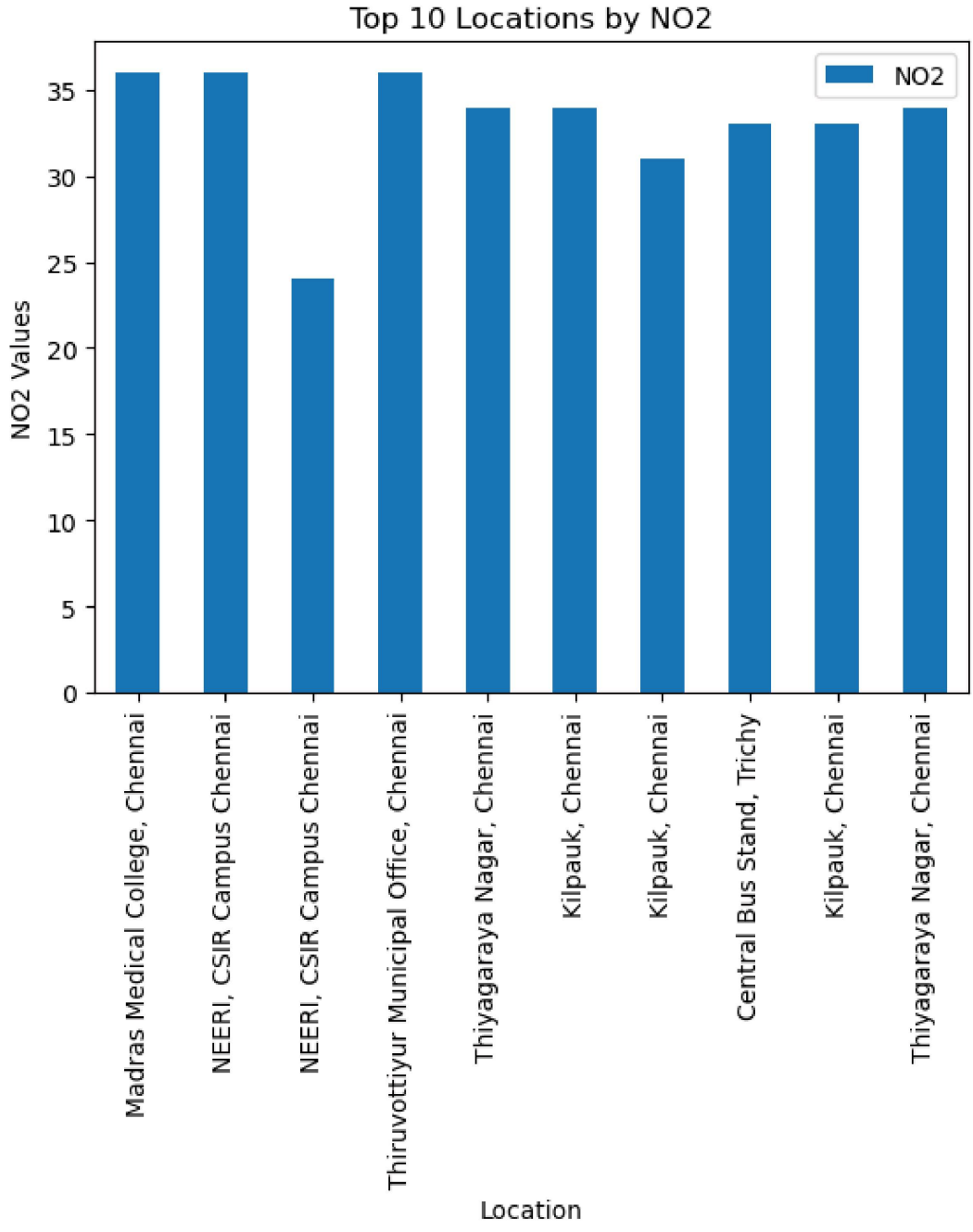


In [ ]:

In [70]:
```
# Assuming 'air_quality_data' is a DataFrame with the relevant columns
loc = dataset[['Location of Monitoring Station', 'SO2', 'NO2', 'RSPM/PM10']]

maxSO2 = loc.sort_values(by='SO2', ascending=False)
maxSO2.head(10).plot(x='Location of Monitoring Station', y='RSPM/PM10', kind='ba

plt.grid(False)
plt.title('Top 10 Locations by RSPM/PM10')
plt.xlabel('Location')
plt.ylabel('RSPM/PM10 Values')
plt.show()
```

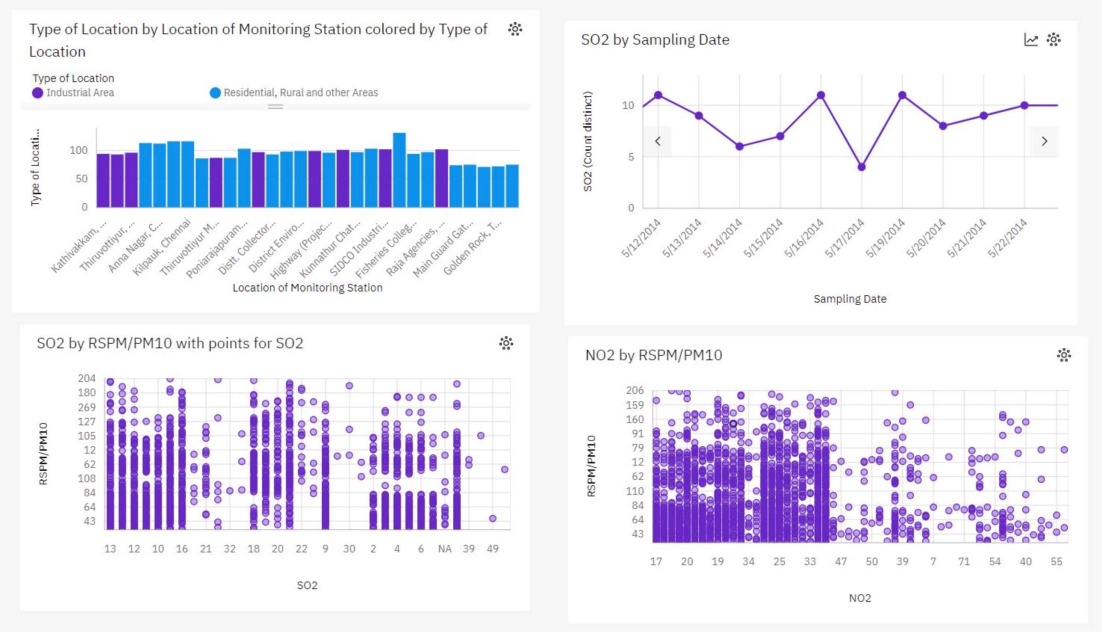


In [69]:
```
# Assuming 'air_quality_data' is a DataFrame with the relevant columns
loc = dataset[['Location of Monitoring Station', 'SO2', 'NO2', 'RSPM/PM10']]

maxSO2 = loc.sort_values(by='SO2', ascending=False)
maxSO2.head(10).plot(x='Location of Monitoring Station', y='NO2', kind='bar')

plt.grid(False)
plt.title('Top 10 Locations by NO2')
plt.xlabel('Location')
plt.ylabel('NO2 Values')
plt.show()
```



In [ ]:

6.1 Data Visualization with IBM Cognos:



Conclusion:

The SVM model outperforms the Decision Tree model in terms of accuracy, with significantly lower mean absolute error and mean squared error. It also has a strong fit to the data, explaining a significant portion of its variance. The Decision Tree model, on the other hand, shows a negative R-squared, indicating a poor fit. Therefore, the SVM model is the better choice for this task, offering superior predictive accuracy and a more robust overall fit to the data.