In [5]:
```
!pip install rasterio
```

```
Collecting rasterio
  Downloading rasterio-1.3.9-cp310-cp310-manylinux2014_x86_64.whl (20.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.6/20.6 MB 75.9 MB/s eta 0:00:00
Collecting affine (from rasterio)
  Downloading affine-2.4.0-py3-none-any.whl (15 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.10/dist-packages (f
rom rasterio) (23.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages
(from rasterio) (2023.7.22)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.10/dist-packag
es (from rasterio) (8.1.7)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packag
es (from rasterio) (0.7.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (f
rom rasterio) (1.23.5)
Collecting snuggs>=1.4.1 (from rasterio)
  Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Requirement already satisfied: click-plugins in /usr/local/lib/python3.10/dist-pac
kages (from rasterio) (1.1.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packag
es (from rasterio) (67.7.2)
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.10/dist-
packages (from snuggs>=1.4.1->rasterio) (3.1.1)
Installing collected packages: snuggs, affine, rasterio
Successfully installed affine-2.4.0 rasterio-1.3.9 snuggs-1.4.7
```

In [1]:
```
import cv2
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call driv
e.mount("/content/drive", force_remount=True).
```

In [2]:
```
# Load the image
img_path= '/content/drive/MyDrive/S5P_NO2_India_2019.png'
img = cv2.imread(img_path)

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Compute the dark channel prior
patch_size = 15
dark_channel = np.zeros_like(gray)
for i in range(gray.shape[0]):
    for j in range(gray.shape[1]):
        patch = gray[max(i-patch_size//2,0):min(i+patch_size//2,gray.shape[0]),
                     max(j-patch_size//2,0):min(j+patch_size//2,gray.shape[1])]
        dark_channel[i,j] = np.min(patch)

# Estimate the atmospheric light
atmosphere = np.percentile(dark_channel, 99)

# Compute the transmission
transmission = 1 - 0.95*dark_channel/atmosphere

# Apply the soft matting algorithm
epsilon = 0.0001
window_size = 15
mean_filter = cv2.blur(transmission, (window_size, window_size))
mean_sqr_filter = cv2.blur(transmission**2, (window_size, window_size))
```

```
variance = mean_sqr_filter - mean_filter**2
a = variance / (variance + epsilon)
a = cv2.blur(a, (window_size, window_size))
transmission_matted = a*transmission + (1-a)*mean_filter

# High-pass filter the image using the transmission map
alpha = 0.1
img_filtered = img - cv2.GaussianBlur(img, (0, 0), np.mean(transmission_matted)*alp

# Add the filtered image to the original image
img_sharp = cv2.add(img, img_filtered)

# Save the sharpened image
cv2.imwrite('sharpened_image.png', img_sharp)
```

Out[2]:
```
True
```

In [3]:
```
import pandas as pd

# Read the data from the CSV file
file_path='/content/drive/MyDrive/2019.xlsx'
df = pd.read_excel(file_path)

# Define a list of cities to consider
cities = ['Visakhapatnam', 'Vijayawada', 'Itanagar', 'Guwahati', 'Patna', 'Muzaffar

# Create an empty DataFrame to store the results
results = pd.DataFrame(columns=['City', 'Annual Average'])

# Loop through each city and calculate the average annual average
for city in cities:
    city_data = df[df['City / town /village'] == city]
    avg_annual_avg = city_data['Annual Average'].mean()
    results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index

# Save the results to a CSV file
results.to_csv('city_average_annual_avg2019.csv', index=False)
```

```
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
```

```
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
```

```
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
```

```
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
```

```
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is depre
cated and will be removed from pandas in a future version. Use pandas.concat inste
ad.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=
True)
```

```
<ipython-input-3-62544a99a1e2>:17: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  results = results.append({'City': city, 'NO_{2}': avg_annual_avg}, ignore_index=True)
```

In [6]:

```python
import pandas as pd
import rasterio
file_path1 = '/content/drive/MyDrive/S5P_NO2_India_2021.tif'

with rasterio.open(file_path1) as src:
    raster_data = src.read(1)

city_df = pd.read_csv('/content/drive/MyDrive/Indian Cities Database.csv')

M = 46.0055 # Molar mass of NO2 in g/mol
V = 0.0002  # Vertical column density of NO2 in mol/m2

for index, row in city_df.iterrows():
    longitude = row['longitude']
    latitude = row['latitude']
    city_name = row['City']
    row, col = src.index(longitude, latitude)
    pixel_value = raster_data[row][col]
    NO2_concentration = (pixel_value * M) / V
    print(f'City: {city_name}, NO2 Concentration: {NO2_concentration} µg/m³')
```

```
City: Abohar, NO2 Concentration: 9.303556346070888 µg/m³
City: Adilabad, NO2 Concentration: 6.876245842827603 µg/m³
City: Agartala, NO2 Concentration: 9.606784465717825 µg/m³
City: Agra, NO2 Concentration: 15.311395666781806 µg/m³
City: Ahmadnagar, NO2 Concentration: 7.713540744758974 µg/m³
City: Ahmedabad, NO2 Concentration: 30.24594167002601 µg/m³
City: Aizawl  , NO2 Concentration: 4.3876709270838035 µg/m³
City: Ajmer, NO2 Concentration: 8.82588256801102 µg/m³
City: Akola, NO2 Concentration: 8.205245565135794 µg/m³
City: Alappuzha, NO2 Concentration: 4.652758416274094 µg/m³
City: Aligarh, NO2 Concentration: 14.455070629317726 µg/m³
City: Alipurduar, NO2 Concentration: 6.598693985000161 µg/m³
City: Allahabad, NO2 Concentration: 13.93029130631717 µg/m³
City: Alwar, NO2 Concentration: 9.610995523354399 µg/m³
City: Ambala, NO2 Concentration: 17.28149601219772 µg/m³
City: Amaravati, NO2 Concentration: 9.769249615739103 µg/m³
City: Amritsar, NO2 Concentration: 18.268919107302267 µg/m³
City: Asansol, NO2 Concentration: 28.119208694076374 µg/m³
City: Aurangabad, NO2 Concentration: 10.346956606253066 µg/m³
City: Aurangabad, NO2 Concentration: 11.522008882960755 µg/m³
City: Bakshpur, NO2 Concentration: 9.030233922192926 µg/m³
City: Bamanpuri, NO2 Concentration: 9.584923088100409 µg/m³
City: Baramula, NO2 Concentration: 4.944329255327467 µg/m³
City: Barddhaman, NO2 Concentration: 14.715516714789576 µg/m³
City: Bareilly, NO2 Concentration: 10.763822908132184 µg/m³
City: Belgaum, NO2 Concentration: 7.299055547298665 µg/m³
City: Bellary, NO2 Concentration: 10.824479244918468 µg/m³
City: Bengaluru, NO2 Concentration: 16.56699940184143 µg/m³
City: Bhagalpur, NO2 Concentration: 12.44128882326275 µg/m³
City: Bharatpur, NO2 Concentration: 9.264903138074441 µg/m³
City: Bharauri, NO2 Concentration: 7.648493400158717 µg/m³
City: Bhatpara, NO2 Concentration: 15.287696799910009 µg/m³
City: Bhavnagar, NO2 Concentration: 8.555392447133679 µg/m³
City: Bhilai, NO2 Concentration: 35.26478576011604 µg/m³
City: Bhilwara, NO2 Concentration: 8.461826201992206 µg/m³
City: Bhiwandi, NO2 Concentration: 14.548263574293374 µg/m³
City: Bhiwani, NO2 Concentration: 12.115967850772112 µg/m³
City: Bhopal , NO2 Concentration: 8.469108229605268 µg/m³
City: Bhubaneshwar, NO2 Concentration: 12.185243722939106 µg/m³
City: Bhuj, NO2 Concentration: 5.439829094832479 µg/m³
City: Bhusaval, NO2 Concentration: 16.984765313635386 µg/m³
City: Bidar, NO2 Concentration: 6.365499871730535 µg/m³
City: Bijapur, NO2 Concentration: 9.051718395492708 µg/m³
City: Bikaner, NO2 Concentration: 7.261158502400334 µg/m³
City: Bilaspur, NO2 Concentration: 22.897730064398505 µg/m³
City: Brahmapur, NO2 Concentration: 6.3412418645340995 µg/m³
City: Budaun, NO2 Concentration: 9.77054077044506 µg/m³
City: Bulandshahr, NO2 Concentration: 17.49491800840664 µg/m³
City: Calicut, NO2 Concentration: 4.635400005630764 µg/m³
City: Chanda, NO2 Concentration: 34.10505584347964 µg/m³
City: Chandigarh , NO2 Concentration: 15.143897491114386 µg/m³
City: Chennai, NO2 Concentration: 14.410727899179077 µg/m³
City: Chikka Mandya, NO2 Concentration: 5.904661409161322 µg/m³
City: Chirala, NO2 Concentration: 4.974305098506 µg/m³
City: Coimbatore, NO2 Concentration: 8.502706004614804 µg/m³
City: Cuddalore, NO2 Concentration: 6.7720595857312365 µg/m³
City: Cuttack, NO2 Concentration: 13.308772755049205 µg/m³
City: Daman, NO2 Concentration: 7.249381808669156 µg/m³
City: Davangere, NO2 Concentration: 6.630422851183536 µg/m³
City: DehraDun, NO2 Concentration: 9.389617974135499 µg/m³
City: Delhi, NO2 Concentration: 43.02833099921156 µg/m³
City: Dhanbad, NO2 Concentration: 19.433594987634823 µg/m³
City: Dibrugarh, NO2 Concentration: 4.061231227330672 µg/m³
City: Dindigul, NO2 Concentration: 6.048854358870503 µg/m³
```

```
City: Dispur, NO2 Concentration: 8.058897768191052 µg/m³
City: Diu, NO2 Concentration: 5.003404791369399 µg/m³
City: Faridabad, NO2 Concentration: 31.09785316142025 µg/m³
City: Firozabad, NO2 Concentration: 21.501933871933563 µg/m³
City: Fyzabad, NO2 Concentration: 10.2047758708038 µg/m³
City: Gangtok, NO2 Concentration: 3.9858284955432355 µg/m³
City: Gaya, NO2 Concentration: 10.300660541801767 µg/m³
City: Ghandinagar, NO2 Concentration: 16.781903391570744 µg/m³
City: Ghaziabad, NO2 Concentration: 35.84519451714243 µg/m³
City: Gopalpur, NO2 Concentration: 12.115023586078534 µg/m³
City: Gulbarga, NO2 Concentration: 9.120116396186903 µg/m³
City: Guntur, NO2 Concentration: 8.127508422134158 µg/m³
City: Gurugram, NO2 Concentration: 23.219734542948835 µg/m³
City: Guwahati, NO2 Concentration: 8.102733766912593 µg/m³
City: Gwalior, NO2 Concentration: 9.65625167834324 µg/m³
City: Haldia, NO2 Concentration: 17.025504808446748 µg/m³
City: Haora, NO2 Concentration: 24.406175688157308 µg/m³
City: Hapur, NO2 Concentration: 14.59522457359631 µg/m³
City: Haripur, NO2 Concentration: 8.376036668877289 µg/m³
City: Hata, NO2 Concentration: 10.568383607667517 µg/m³
City: Hindupur, NO2 Concentration: 5.772808076893882 µg/m³
City: Hisar, NO2 Concentration: 12.030461574379308 µg/m³
City: Hospet, NO2 Concentration: 13.354077741946453 µg/m³
City: Hubli, NO2 Concentration: 6.940831741064935 µg/m³
City: Hyderabad, NO2 Concentration: 19.88240035167207 µg/m³
City: Imphal, NO2 Concentration: 4.973327617644283 µg/m³
City: Indore, NO2 Concentration: 11.775606661841225 µg/m³
City: Itanagar, NO2 Concentration: 3.924471203609077 µg/m³
City: Jabalpur, NO2 Concentration: 8.094756389480178 µg/m³
City: Jaipur, NO2 Concentration: 15.40643795601386 µg/m³
City: Jammu, NO2 Concentration: 11.134721001923605 µg/m³
City: Jamshedpur, NO2 Concentration: 36.756997297489875 µg/m³
City: Jhansi, NO2 Concentration: 9.240541384655824 µg/m³
City: Jodhpur, NO2 Concentration: 10.783214536006133 µg/m³
City: Jorhat, NO2 Concentration: 5.584498460772537 µg/m³
City: Kagaznagar, NO2 Concentration: 9.953719409940238 µg/m³
City: Kakinada, NO2 Concentration: 7.416375865470754 µg/m³
City: Kalyan, NO2 Concentration: 15.600602532744333 µg/m³
City: Karimnagar, NO2 Concentration: 10.732038485234098 µg/m³
City: Karnal, NO2 Concentration: 14.344769804566202 µg/m³
City: Karur, NO2 Concentration: 7.091835220293401 µg/m³
City: Kavaratti, NO2 Concentration: 2.4761517052123407 µg/m³
City: Khammam, NO2 Concentration: 8.58382208012436 µg/m³
City: Khanapur, NO2 Concentration: 6.193029064041996 µg/m³
City: Kochi, NO2 Concentration: 7.40031896803004 µg/m³
City: Kohima, NO2 Concentration: 4.136634890377544 µg/m³
City: Kolar, NO2 Concentration: 5.214609228549364 µg/m³
City: Kolhapur, NO2 Concentration: 10.451174534201767 µg/m³
City: Kolkata , NO2 Concentration: 25.999951255574963 µg/m³
City: Kollam, NO2 Concentration: 4.30830973453968 µg/m³
City: Kota, NO2 Concentration: 19.987360553550925 µg/m³
City: Krishnanagar, NO2 Concentration: 10.781889533866922 µg/m³
City: Krishnapuram, NO2 Concentration: 6.565145755531653 µg/m³
City: Kumbakonam, NO2 Concentration: 6.2281826769499835 µg/m³
City: Kurnool, NO2 Concentration: 8.38471170255134 µg/m³
City: Latur, NO2 Concentration: 6.7990345606354925 µg/m³
City: Lucknow, NO2 Concentration: 14.094985608723134 µg/m³
City: Ludhiana, NO2 Concentration: 21.111370865329746 µg/m³
City: Machilipatnam, NO2 Concentration: 4.691129423121771 µg/m³
City: Madurai, NO2 Concentration: 7.597498746376209 µg/m³
City: Mahabubnagar, NO2 Concentration: 7.108291031607716 µg/m³
City: Malegaon Camp, NO2 Concentration: 7.9614198075464975 µg/m³
City: Mangalore, NO2 Concentration: 6.800285380354956 µg/m³
City: Mathura, NO2 Concentration: 12.441459253714712 µg/m³
```

```
City: Meerut, NO2 Concentration: 15.28352013403428 µg/m³
City: Mirzapur, NO2 Concentration: 11.255717851774046 µg/m³
City: Moradabad, NO2 Concentration: 12.386489530388927 µg/m³
City: Mumbai, NO2 Concentration: 27.2599293293989 µg/m³
City: Muzaffarnagar, NO2 Concentration: 13.09637369643862 µg/m³
City: Muzaffarpur, NO2 Concentration: 13.776046623471474 µg/m³
City: Mysore, NO2 Concentration: 7.5384506041992205 µg/m³
City: Nagercoil, NO2 Concentration: 3.429286262635481 µg/m³
City: Nalgonda, NO2 Concentration: 6.401353212976479 µg/m³
City: Nanded, NO2 Concentration: 7.740113250113345 µg/m³
City: Nandyal, NO2 Concentration: 6.65675470368617 µg/m³
City: Nasik, NO2 Concentration: 12.379251996826302 µg/m³
City: Navsari, NO2 Concentration: 8.320957857707919 µg/m³
City: Nellore, NO2 Concentration: 9.938107673761017 µg/m³
City: New Delhi, NO2 Concentration: 36.997214820230994 µg/m³
City: Nizamabad, NO2 Concentration: 7.880523101758766 µg/m³
City: Ongole, NO2 Concentration: 5.061741093300289 µg/m³
City: Pali, NO2 Concentration: 6.830516341321107 µg/m³
City: Panaji, NO2 Concentration: 5.86610422079079 µg/m³
City: Panchkula, NO2 Concentration: 15.583007902697354 µg/m³
City: Panipat, NO2 Concentration: 26.32396978863321 µg/m³
City: Parbhani, NO2 Concentration: 6.906058067080982 µg/m³
City: Pathankot, NO2 Concentration: 8.372176728604138 µg/m³
City: Patiala, NO2 Concentration: 14.084954704435404 µg/m³
City: Patna, NO2 Concentration: 13.28096528937855 µg/m³
City: Pilibhit, NO2 Concentration: 8.373507792778176 µg/m³
City: Porbandar, NO2 Concentration: 6.035141095751536 µg/m³
City: Port Blair, NO2 Concentration: 2.3530815183181737 µg/m³
City: Proddatur, NO2 Concentration: 7.946779741709849 µg/m³
City: Puducherry, NO2 Concentration: 5.820443393994466 µg/m³
City: Pune, NO2 Concentration: 16.716668593169533 µg/m³
City: Puri, NO2 Concentration: 6.435603727223444 µg/m³
City: Purnea, NO2 Concentration: 8.691106216825792 µg/m³
City: Raichur, NO2 Concentration: 10.029151488567175 µg/m³
City: Raipur, NO2 Concentration: 28.670297438039988 µg/m³
City: Rajahmundry, NO2 Concentration: 8.590324807221315 µg/m³
City: Rajapalaiyam, NO2 Concentration: 4.781643083582 µg/m³
City: Rajkot, NO2 Concentration: 12.137493784692523 µg/m³
City: Ramagundam, NO2 Concentration: 10.522388563289889 µg/m³
City: Rampura, NO2 Concentration: 15.498166366428485 µg/m³
City: Ranchi, NO2 Concentration: 10.298714564727502 µg/m³
City: Ratlam, NO2 Concentration: 6.804218536412837 µg/m³
City: Raurkela, NO2 Concentration: 26.180073617612003 µg/m³
City: Rohtak, NO2 Concentration: 13.279273041989846 µg/m³
City: Saharanpur, NO2 Concentration: 13.572560875103617 µg/m³
City: Saidapur, NO2 Concentration: 8.839291669647285 µg/m³
City: Saidpur, NO2 Concentration: 4.64879229336991 µg/m³
City: Salem, NO2 Concentration: 8.124001528187584 µg/m³
City: Samlaipadar, NO2 Concentration: 18.890799748525172 µg/m³
City: Sangli, NO2 Concentration: 9.283282466175471 µg/m³
City: Saugor, NO2 Concentration: 6.722287751908369 µg/m³
City: Shahbazpur, NO2 Concentration: 14.1807211893796 µg/m³
City: Shiliguri, NO2 Concentration: 8.968029335482676 µg/m³
City: Shillong , NO2 Concentration: 5.458188879743544 µg/m³
City: Shimla, NO2 Concentration: 7.342204889048742 µg/m³
City: Shimoga, NO2 Concentration: 5.361811941484227 µg/m³
City: Sikar, NO2 Concentration: 6.7088596644817695 µg/m³
City: Silchar, NO2 Concentration: 5.152665971965583 µg/m³
City: Silvassa, NO2 Concentration: 9.014479748940724 µg/m³
City: Sirsa, NO2 Concentration: 9.82275576835854 µg/m³
City: Sonipat, NO2 Concentration: 14.255652736552477 µg/m³
City: Srinagar, NO2 Concentration: 8.65536961134591 µg/m³
City: Surat, NO2 Concentration: 17.290426615088116 µg/m³
City: Tezpur, NO2 Concentration: 6.159982731491605 µg/m³
```

```
City: Thanjavur, NO2 Concentration: 6.139151272897391 µg/m³
City: Tharati Etawah, NO2 Concentration: 9.5784754473401 µg/m³
City: Thiruvananthapuram, NO2 Concentration: 4.379028998174555 µg/m³
City: Tiruchchirappalli, NO2 Concentration: 7.741876420858871 µg/m³
City: Tirunelveli, NO2 Concentration: 4.625301047210951 µg/m³
City: Tirupati, NO2 Concentration: 6.707237517654729 µg/m³
City: Tiruvannamalai, NO2 Concentration: 5.57110219819094 µg/m³
City: Tonk, NO2 Concentration: 7.043835081482421 µg/m³
City: Tuticorin, NO2 Concentration: 9.416151488811565 µg/m³
City: Udaipur, NO2 Concentration: 9.099541471395733 µg/m³
City: Ujjain, NO2 Concentration: 7.4962568151937266 µg/m³
City: Vadodara, NO2 Concentration: 16.52029901727529 µg/m³
City: Valparai, NO2 Concentration: 3.7328862656661723 µg/m³
City: Varanasi, NO2 Concentration: 14.79690488953124 µg/m³
City: Vellore, NO2 Concentration: 6.1439910234804165 µg/m³
City: Vishakhapatnam, NO2 Concentration: 16.276269762599522 µg/m³
City: Vizianagaram, NO2 Concentration: 7.8124057954448025 µg/m³
City: Warangal, NO2 Concentration: 8.480857882641876 µg/m³
City: Jorapokhar, NO2 Concentration: 22.32062851826693 µg/m³
City: Brajrajnagar, NO2 Concentration: 30.606760287791744 µg/m³
City: Talcher, NO2 Concentration: 42.266383102442056 µg/m³
```

In [7]:

```
import pandas as pd
import rasterio

# Load air quality station data
station_data = pd.read_csv('/content/drive/MyDrive/Ind_SW_NO2.csv',encoding= 'unicc

# Load Sentinel data
file_path = '/content/drive/MyDrive/S5P_NO2_India_2019.tif'
with rasterio.open(file_path) as src:
    raster_data = src.read(1)

# Convert Sentinel pixel values to NO2 concentration
M = 46.0055 # Molar mass of NO2 in g/mol
V = 0.0002  # Vertical column density of NO2 in mol/m2
city_data = []
for index, row in station_data.iterrows():
    longitude = row['Longitude']
    latitude = row['Latitude']
    city_name = row['City']
    row, col = src.index(longitude, latitude)
    pixel_value = raster_data[row][col]
    NO2_concentration = (pixel_value * M) / V
    city_data.append([city_name, NO2_concentration])

# Create DataFrame from Sentinel data
sentinel_data = pd.DataFrame(city_data, columns=['City', 'Sentinel NO2 Reading_2019

# Merge air quality station data and Sentinel data
merged_data = pd.merge(station_data, sentinel_data, on='City')

# Calculate correlation between air quality station NO2 readings and Sentinel NO2 r
correlation = merged_data['NO2 Reading_2019 (µg/m³)'].corr(merged_data['Sentinel NC

# Print correlation
print(f"Correlation between air quality station data and Sentinel data: {correlatic
```

```
Correlation between air quality station data and Sentinel data: 0.5332283650110543
```

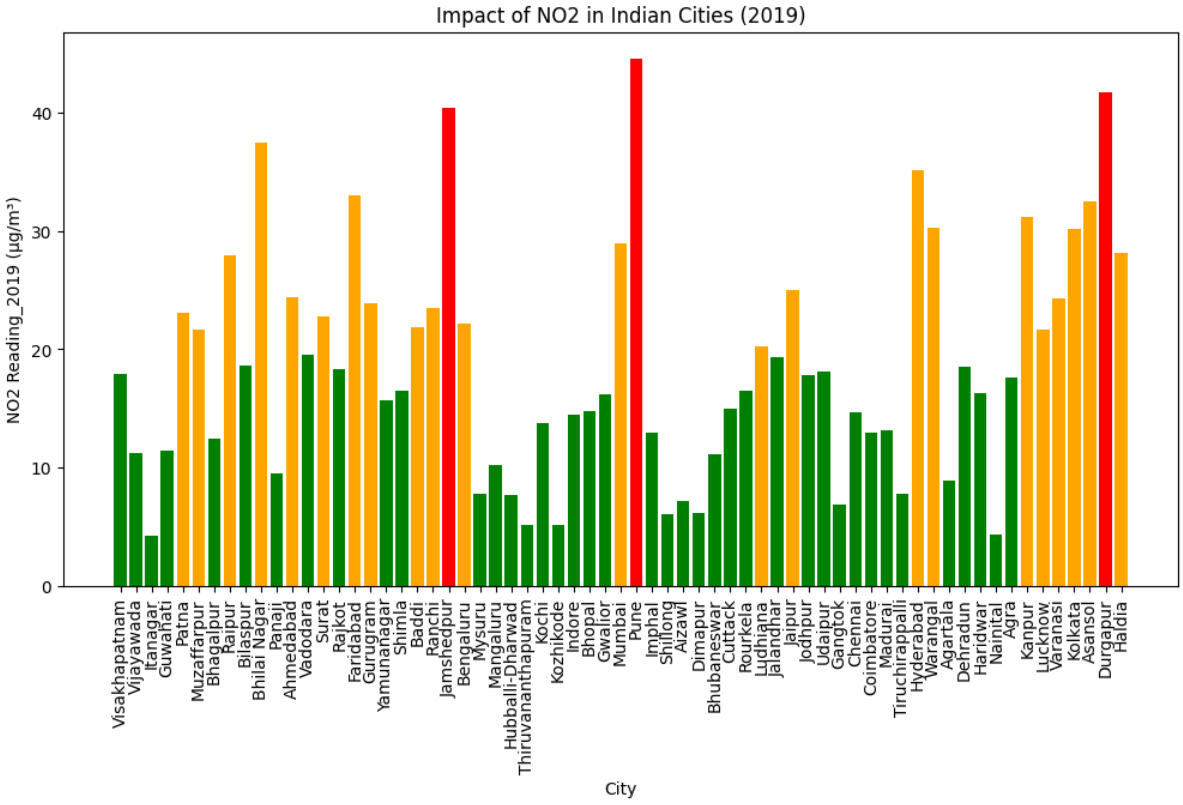
In [8]:

```
import pandas as pd
import rasterio

# Load air quality station data
```

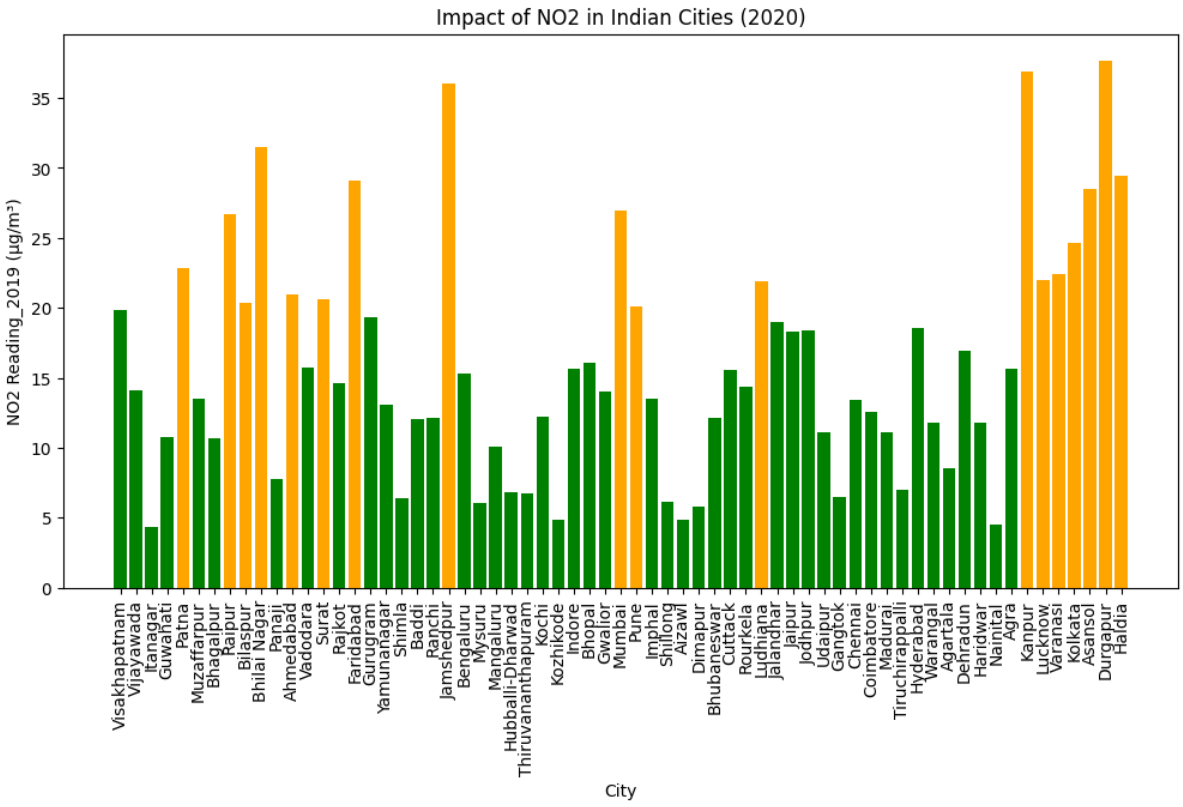
```
station_data = pd.read_csv('/content/drive/MyDrive/Ind_SW_NO2.csv',encoding= 'unicc
station_data = station_data[['City', 'NO2 Reading_2019 (µg/m³)', 'Latitude', 'Longi

# Load Sentinel data
file_path = '/content/drive/MyDrive/S5P_NO2_India_2019.tif'
with rasterio.open(file_path) as src:
    raster_data = src.read(1)

M = 46.0055 # Molar mass of NO2 in g/mol
V = 0.0002  # Vertical column density of NO2 in mol/m2

# Combine air quality station data with Sentinel data
for index, row in station_data.iterrows():
    city_name = row['City']
    latitude = row['Latitude']
    longitude = row['Longitude']
    row, col = src.index(longitude, latitude)
    pixel_value = raster_data[row][col]
    NO2_concentration = (pixel_value * M) / V
    station_NO2 = station_data.loc[index, 'NO2 Reading_2019 (µg/m³)']
    combined_NO2 = (station_NO2 + NO2_concentration) / 2
    station_data.at[index, 'NO2 Reading_2019 (µg/m³)'] = combined_NO2

# Save the combined data to a new file
station_data.to_excel('Combined_NO2_Data_2019.xlsx', index=False)
```

In [9]:

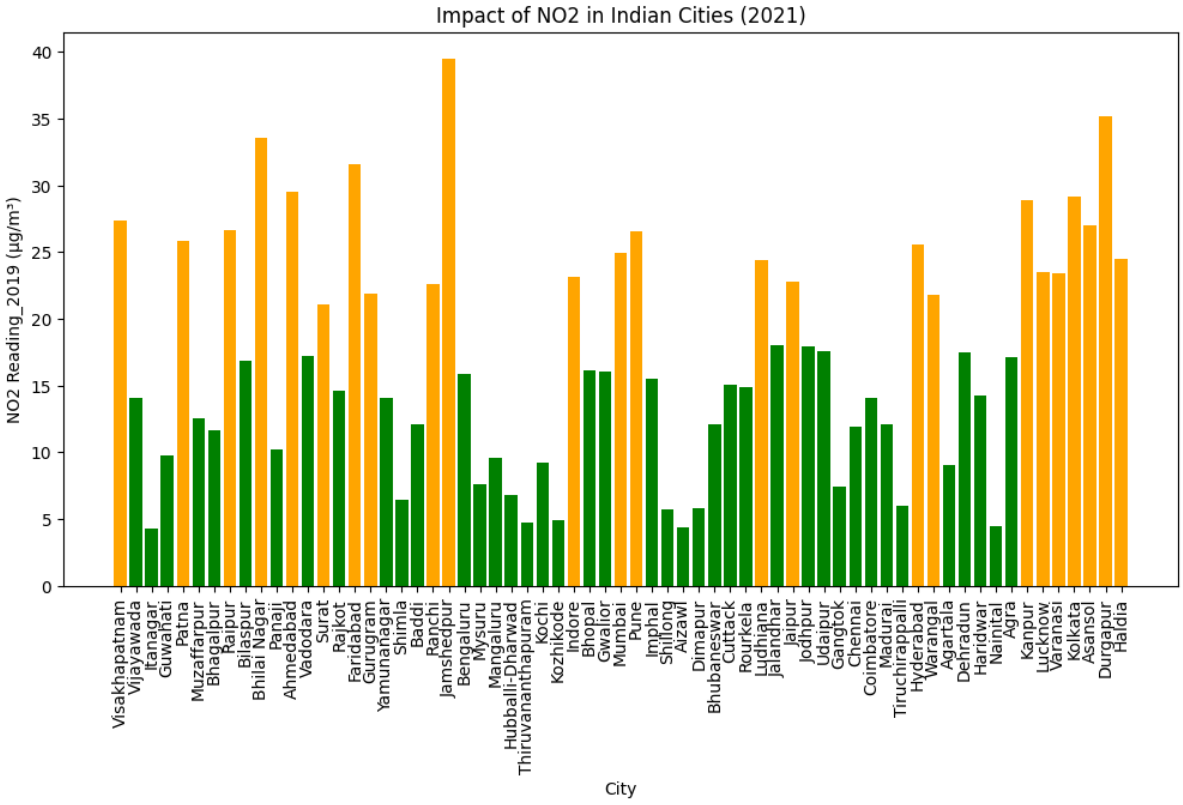
```
import pandas as pd
import matplotlib.pyplot as plt

# Load air quality station data
station_data = pd.read_excel('Combined_NO2_Data_2019.xlsx')
station_data = station_data[['City', 'NO2 Reading_2019 (µg/m³)', 'Latitude', 'Longi

# Define impact levels based on NO2 readings
...


# Define impact levels based on NO2 readings
def impact_level(value):
    if value >= 40:
        return 'Highly impacted'
    elif value >= 20:
        return 'Moderately impacted'
    else:
        return 'Low impacted'

# Add a new column for impact level
station_data['Impact level'] = station_data['NO2 Reading_2019 (µg/m³)'].apply(impac

# Plot the impact levels by city
plt.figure(figsize=(12,6))
plt.bar(station_data['City'], station_data['NO2 Reading_2019 (µg/m³)'], color=stati
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('NO2 Reading_2019 (µg/m³)')
plt.title('Impact of NO2 in Indian Cities (2019)')
plt.show()
```



In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load air quality station data
station_data = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2020.xlsx')
station_data = station_data[['City', 'NO2 Reading (µg/m³)', 'Latitude', 'Longitude'


# Define impact levels based on NO2 readings
def impact_level(value):
    if value >= 40:
        return 'Highly impacted'
    elif value >= 20:
        return 'Moderately impacted'
    else:
        return 'Low impacted'

# Add a new column for impact level
station_data['Impact level'] = station_data['NO2 Reading (µg/m³)'].apply(impact_lev

# Plot the impact levels by city
plt.figure(figsize=(12,6))
plt.bar(station_data['City'], station_data['NO2 Reading (µg/m³)'], color=station_da
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('NO2 Reading_2019 (µg/m³)')
plt.title('Impact of NO2 in Indian Cities (2020)')
plt.show()
```



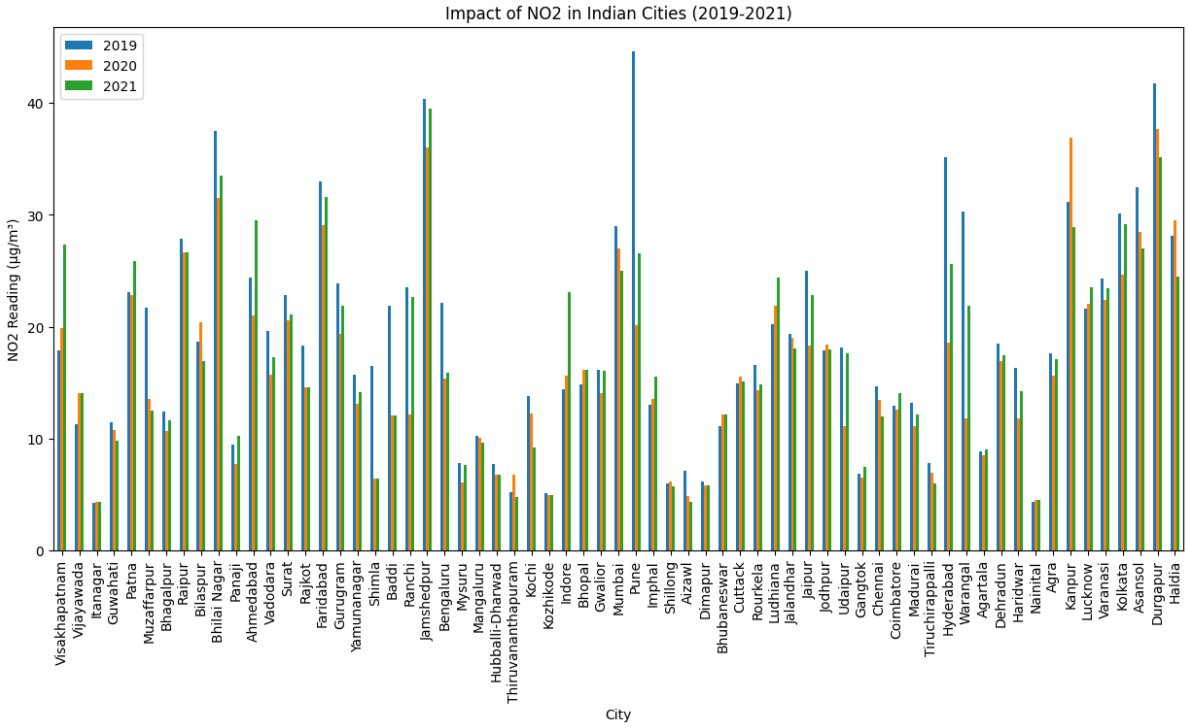
In [11]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load air quality station data
station_data = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2021.xlsx')
station_data = station_data[['City', 'NO2 Reading_2021 (µg/m³)', 'Latitude', 'Longi


# Define impact levels based on NO2 readings
def impact_level(value):
    if value >= 40:
        return 'Highly impacted'
    elif value >= 20:
        return 'Moderately impacted'
    else:
        return 'Low impacted'

# Add a new column for impact level
station_data['Impact level'] = station_data['NO2 Reading_2021 (µg/m³)'].apply(impac

# Plot the impact levels by city
plt.figure(figsize=(12,6))
plt.bar(station_data['City'], station_data['NO2 Reading_2021 (µg/m³)'], color=stati
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('NO2 Reading_2019 (µg/m³)')
plt.title('Impact of NO2 in Indian Cities (2021)')
plt.show()
```



In [12]:

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

# Load air quality station data for each year
data_2019 = pd.read_excel('Combined_NO2_Data_2019.xlsx')
data_2020 = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2020.xlsx')
data_2021 = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2021.xlsx')

# Merge data for all years into one dataframe
data = pd.merge(data_2019, data_2020, on='City', suffixes=('_2019', '_2020'))
data = pd.merge(data, data_2021, on='City', suffixes=('','_2021'))

# Select necessary columns
data = data[['City', 'NO2 Reading_2019 (µg/m³)', 'NO2 Reading (µg/m³)', 'NO2 Readir

# Define impact levels based on NO2 readings
def impact_level(value):
    if value >= 40:
        return 'Highly impacted'
    elif value >= 20:
        return 'Moderately impacted'
    else:
        return 'Low impacted'

# Add impact level columns for each year
data['Impact level_2019'] = data['NO2 Reading_2019 (µg/m³)'].apply(impact_level)
data['Impact level_2020'] = data['NO2 Reading (µg/m³)'].apply(impact_level)
data['Impact level_2021'] = data['NO2 Reading_2021 (µg/m³)'].apply(impact_level)

# Set city as index
data = data.set_index('City')

# Plot a grouped bar chart for each year
data.plot(kind='bar', figsize=(15, 7))
plt.xlabel('City')
plt.ylabel('NO2 Reading (µg/m³)')
plt.title('Impact of NO2 in Indian Cities (2019-2021)')
```

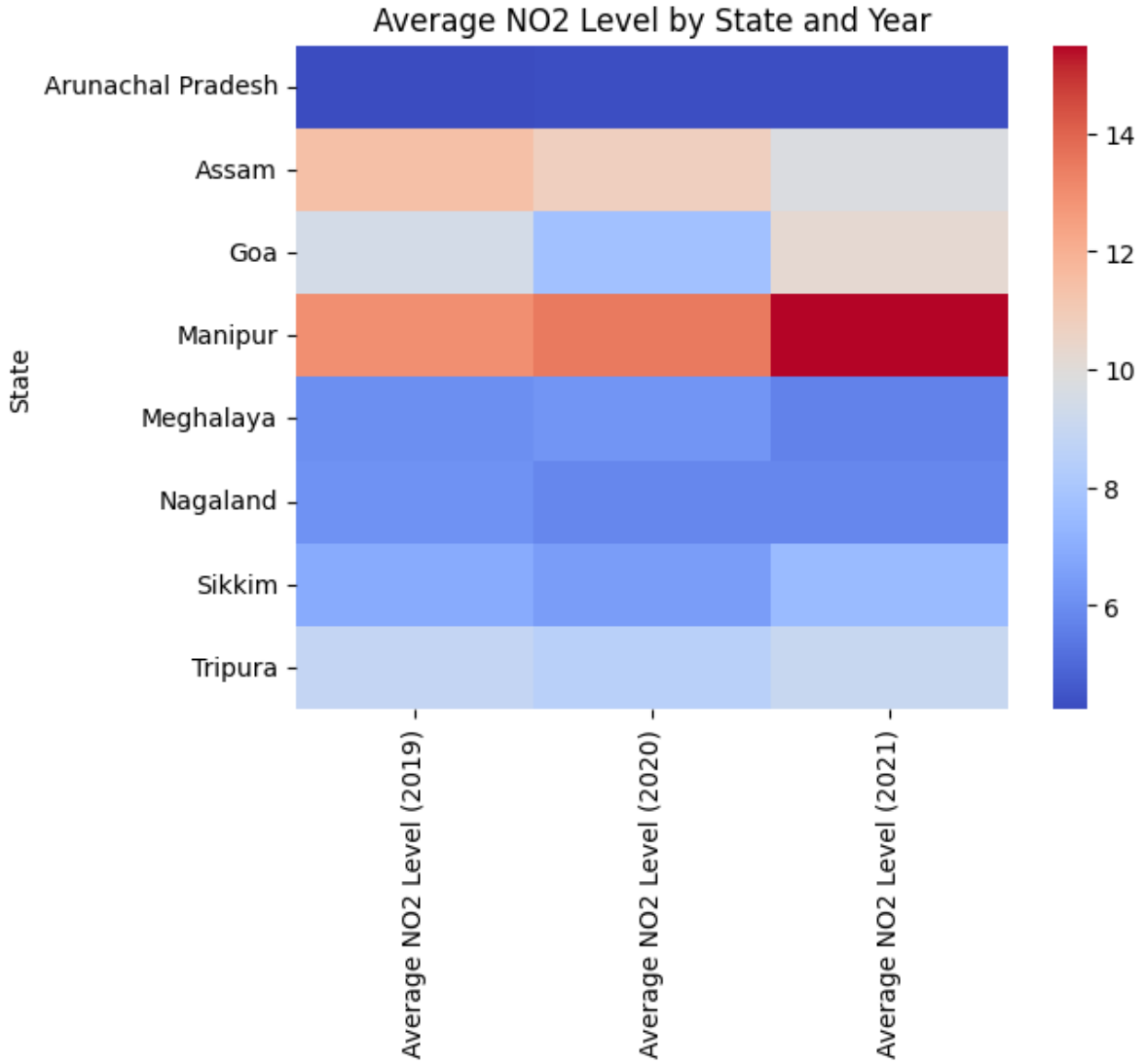
```
plt.legend(['2019', '2020', '2021'])
plt.show()
```



In [13]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load state-wise city data
state_data = pd.read_csv('/content/drive/MyDrive/State_City.csv', encoding='ISO-885

# Load air quality station data
aq_data_2019 = pd.read_excel('Combined_NO2_Data_2019.xlsx')
aq_data_2020 = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2020.xlsx')
aq_data_2021 = pd.read_excel('/content/drive/MyDrive/Combined_NO2_Data_2021.xlsx')

# Merge air quality station data with state-wise city data
aq_data_2019 = pd.merge(aq_data_2019, state_data, how='left', left_on='City', right
aq_data_2020 = pd.merge(aq_data_2020, state_data, how='left', left_on='City', right
aq_data_2021 = pd.merge(aq_data_2021, state_data, how='left', left_on='City', right

# Define impact levels based on NO2 readings
def impact_level(value):
    if value >= 40:
        return 'Highly impacted'
    elif value >= 20:
        return 'Moderately impacted'
    else:
        return 'Low impacted'
# Group the data by state and year to get the average NO2 level for each state in e
avg_aq_data_2019 = aq_data_2019.groupby(['State'])['NO2 Reading_2019 (µg/m³)'].mean
avg_aq_data_2020 = aq_data_2020.groupby(['State'])['NO2 Reading (µg/m³)'].mean().re
avg_aq_data_2021 = aq_data_2021.groupby(['State'])['NO2 Reading_2021 (µg/m³)'].mean
avg_aq_data = pd.merge(avg_aq_data_2019, avg_aq_data_2020, on='State')
avg_aq_data = pd.merge(avg_aq_data, avg_aq_data_2021, on='State')

# Visualize the data using a heat map
sns.heatmap(avg_aq_data.set_index('State'), cmap='coolwarm')
plt.title('Average NO2 Level by State and Year')
plt.show()
```

```
# Find the highest and lowest average NO2 level for each state across the three yea
highest_avg = avg_aq_data.set_index('State').max(axis=1)
lowest_avg = avg_aq_data.set_index('State').min(axis=1)
print('Highest average NO2 level by state:')
print(highest_avg)
print('\nLowest average NO2 level by state:')
print(lowest_avg)
```



```
Highest average NO2 level by state:
State
Arunachal Pradesh     4.317776
Assam                11.446099
Goa                  10.230718
Manipur              15.498955
Meghalaya             6.177768
Nagaland              6.125036
Sikkim                7.467213
Tripura               9.002322
dtype: float64

Lowest average NO2 level by state:
State
Arunachal Pradesh     4.238407
Assam                 9.773614
Goa                   7.730718
Manipur              12.971575
Meghalaya             5.677768
Nagaland              5.812471
Sikkim                6.467213
Tripura               8.502322
dtype: float64
```

In [ ]: