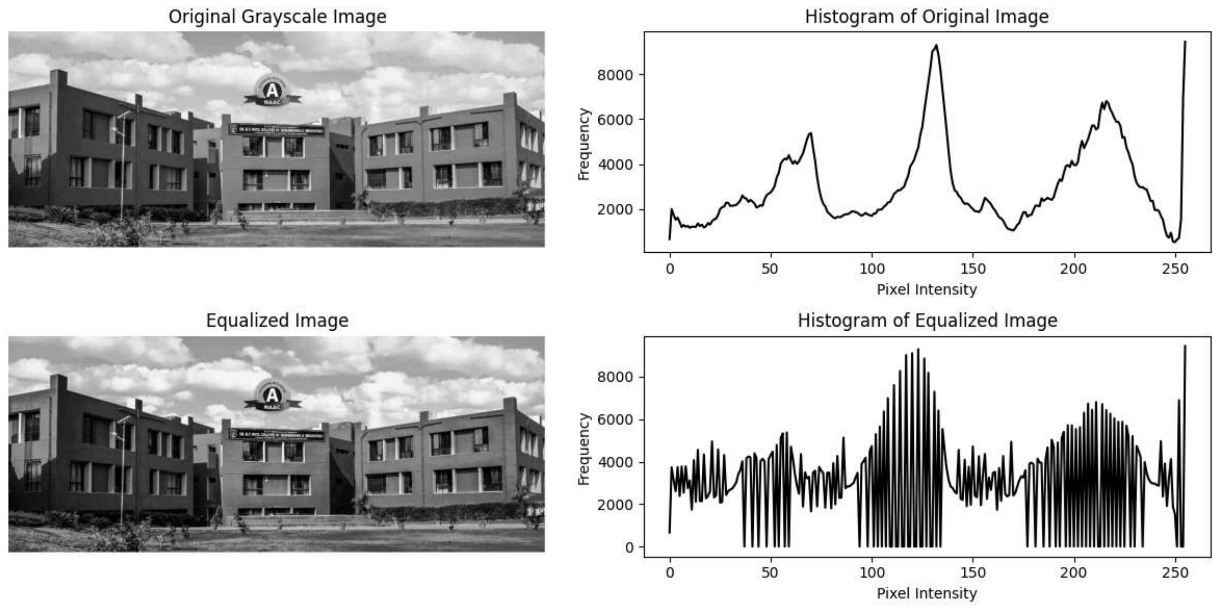
Practical No. 3

```
In [3]:
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Read and Load the Image
image = cv2.imread('image.jpg')  # Replace with the path to your image

# Step 2: Convert to Grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Step 3: Compute Histogram of the Original Image
hist_original = cv2.calcHist([gray_image], [0], None, [256], [0, 256])

# Step 4: Apply Histogram Equalization
equalized_image = cv2.equalizeHist(gray_image)

# Step 5: Compute Histogram of the Equalized Image
hist_equalized = cv2.calcHist([equalized_image], [0], None, [256], [0, 256])

# Step 6: Display Images and Histograms
plt.figure(figsize=(12, 6))

# Display Original Image
plt.subplot(2, 2, 1)
plt.imshow(gray_image, cmap='gray')
plt.title('Original Grayscale Image')
plt.axis('off')

# Display Histogram of Original Image
plt.subplot(2, 2, 2)
plt.plot(hist_original, color='black')
plt.title('Histogram of Original Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Display Equalized Image
plt.subplot(2, 2, 3)
plt.imshow(equalized_image, cmap='gray')
plt.title('Equalized Image')
plt.axis('off')

# Display Histogram of Equalized Image
plt.subplot(2, 2, 4)
plt.plot(hist_equalized, color='black')
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```


Original Grayscale Image
Histogram of Original Image
Frequency
Pixel Intensity
Equalized Image
Histogram of Equalized Image
Frequency
Pixel Intensity

In [ ]: