**IMPLEMENTING PROGRAM FOR VISUALIZING TIME SERIES DATA**

**AIM:**

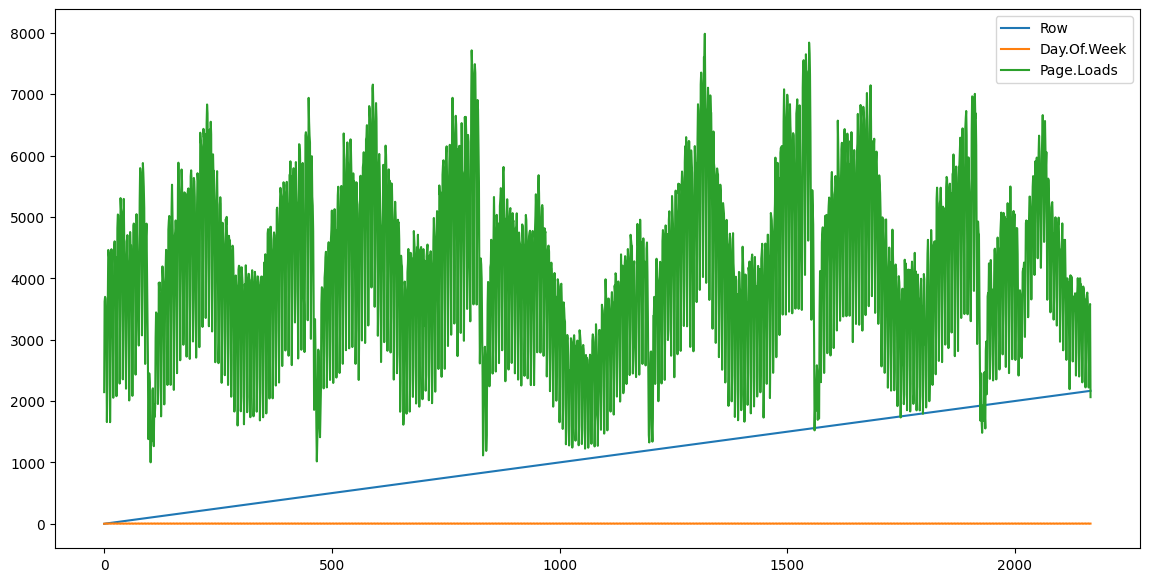
Implement programs for visualizing time series data

**PROCEDURE:**

data = df.drop\_duplicates()

print(f"Dataset now has {data.shape[0]} rows and {data.shape[1]} columns.")

df.plot(figsize=(14,7))



df = df.dropna()

df.shape

plt.figure(figsize=(14, 7))

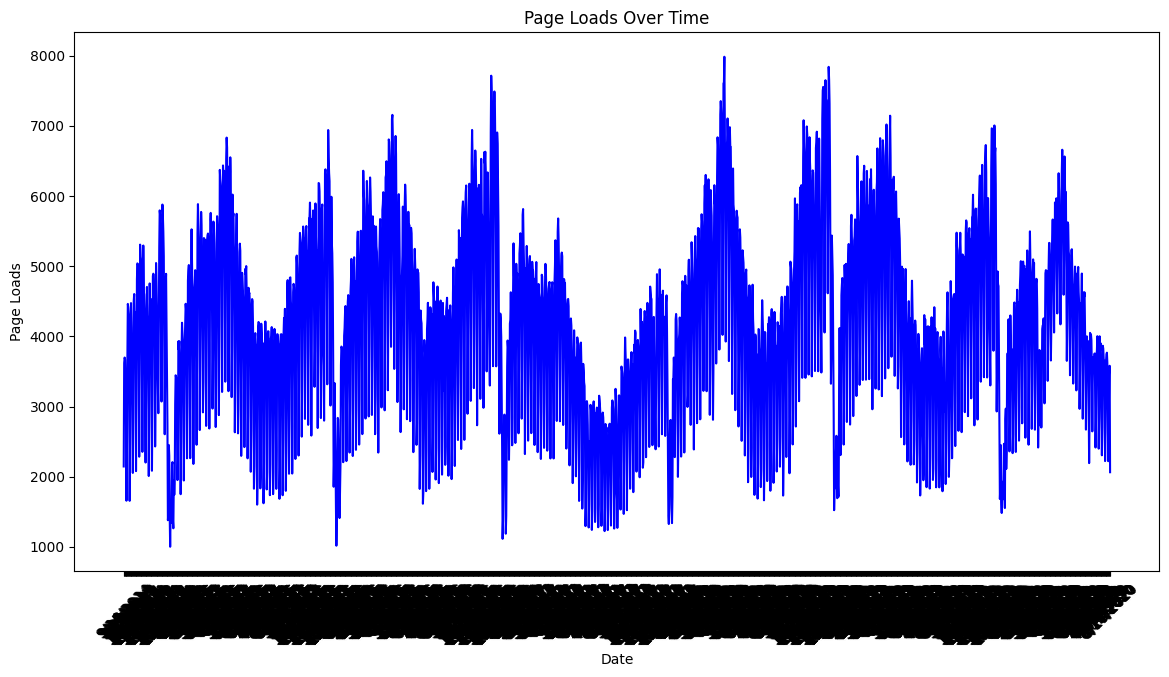
plt.plot(df['Date'], df['Page.Loads'], color='blue')

plt.title('Page Loads Over Time')

plt.xlabel('Date')

plt.ylabel('Page Loads')

plt.xticks(rotation=45)

plt.show()

plt.figure(figsize=(8, 5))

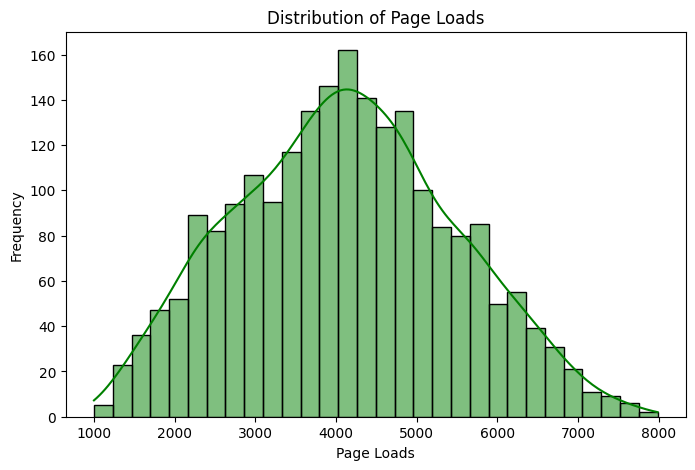
sns.histplot(df['Page.Loads'], bins=30, kde=True, color='green')

plt.title('Distribution of Page Loads')

plt.xlabel('Page Loads')

plt.ylabel('Frequency')

plt.show()



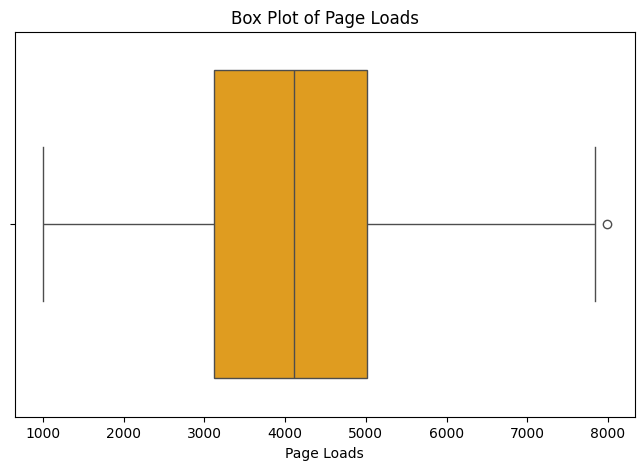
plt.figure(figsize=(8, 5))

sns.boxplot(x=df['Page.Loads'], color='orange')

plt.title('Box Plot of Page Loads')

plt.xlabel('Page Loads')

plt.show()



plt.figure(figsize=(8, 5))

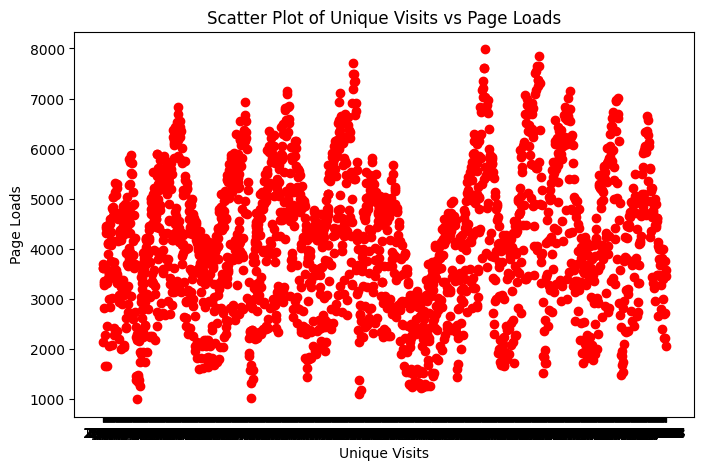
plt.scatter(df['Unique.Visits'], df['Page.Loads'], color='red')

plt.title('Scatter Plot of Unique Visits vs Page Loads')

plt.xlabel('Unique Visits')

plt.ylabel('Page Loads')

plt.show()



rolling\_mean = df['Page.Loads'].rolling(window=7).mean()

rolling\_std = df['Page.Loads'].rolling(window=7).std()

plt.figure(figsize=(14, 7))

plt.plot(df['Page.Loads'], color='blue', label='Original')

plt.plot(rolling\_mean, color='red', label='Rolling Mean')

plt.plot(rolling\_std, color='black', label='Rolling Std')

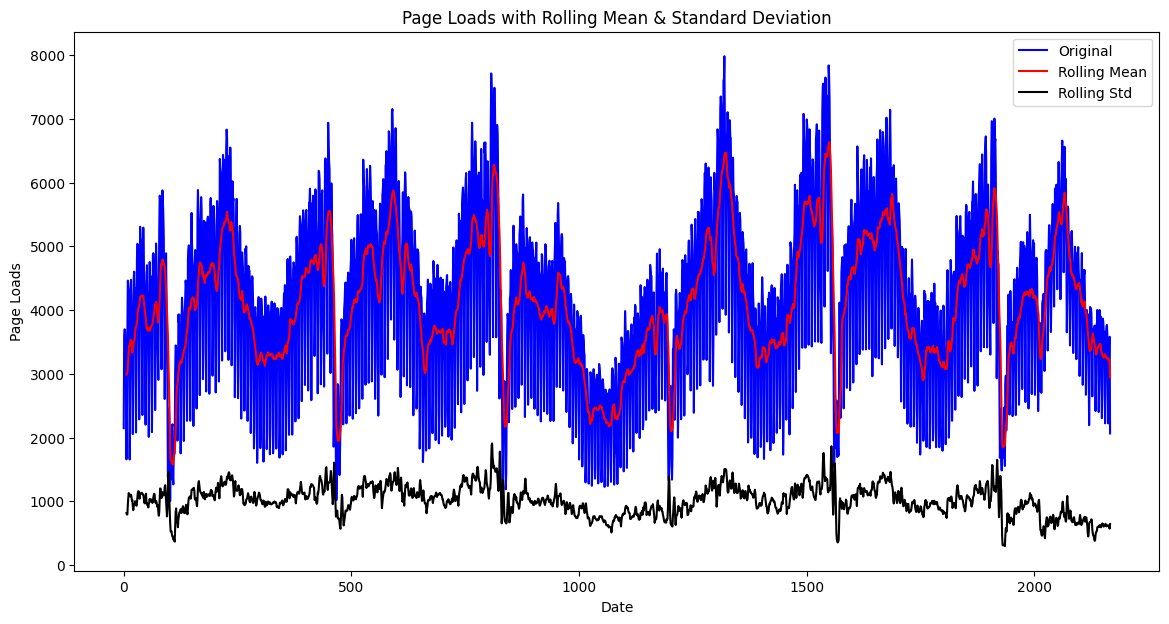
plt.title('Page Loads with Rolling Mean & Standard Deviation')

plt.xlabel('Date')

plt.ylabel('Page Loads')

plt.legend()

plt.show()



**RESULT:**

Thus the program has bees excecuted successfully.