Introduction to Facebook Analysis

In today's digital age, effective social media management is paramount for businesses and organizations looking to connect with their audience and enhance their online presence. Harnessing the power of platforms like Facebook, understanding audience behavior, and optimizing content strategy can be a game-changer. To this end, I embarked on a comprehensive social media analysis, primarily focusing on Facebook.

Our analysis delves into a plethora of facets, from dissecting engagement metrics to exploring content types and evaluating time-based trends. By harnessing data-driven insights, we aim to provide a clear picture of what works best on this social media platform and offer actionable recommendations for maximizing engagement, reach, and overall impact.

The journey begins with an examination of key engagement metrics, such as likes, comments, and shares, to gauge the audience's interaction with posts. We dive deep into content types, unraveling the performance of text, photo, and video content, and shed light on which resonates most with the audience.

Time is a critical dimension in social media strategy, and our analysis dissects how engagement and reach vary throughout the day and across days of the week. Insights into peak engagement times and high-impact days can inform scheduling strategies for optimal results.

Moreover, we explore monthly trends, unearthing patterns in user activity and content performance over the course of a year. By identifying peaks and troughs in engagement, we aim to help tailor content strategy to capitalize on seasonal variations.

Yearly trends provide an overarching view of how key metrics have evolved over time. We examine the impact of significant events or shifts in user behavior, such as the influence of external factors like the COVID-19 pandemic on social media engagement.

This multifaceted social media analysis combines quantitative data with qualitative insights, offering a holistic understanding of the bank's Facebook presence. It equips decision-makers with the knowledge to refine content strategies, optimize posting schedules, and ultimately enhance their social media impact.

Join me on this data-driven journey as wI navigate through the vast landscape of social media analytics, unearthing valuable insights that can transform your social media strategy.

In [1]:
```python
#importing important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas import set_option
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import acf, pacf
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from pmdarima import auto_arima
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
from prophet import Prophet
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
data = 'C:\\Users\\HP\\Documents\\WORKSPACE\\Post Performance (Stanbic IBTC) January 1,
df = pd.read_excel(data)
```

In [226]:
```python
#exploring the data
df.head()
```

In [3]:
```python
# Set pandas options to display all columns
pd.set_option('display.max_columns', None)
```

Out[226]:

	Date	Post ID	Network	Post Type	Content Type	Profile	Sent by	
0	12/17/2022	253788558082460_544102044602588б	Facebook	Post	Video	Stanbic	Damilare	https://www.

In [2]:
```
data = 'C:\\Users\\HP\\Documents\\WORKSPACE\\Post Performance (Stanbic IBTC) January 1,
df = pd.read_excel(data)
```

In [226]:
```
#exploring the data
df.head()
```

In [3]:
```
# Set pandas options to display all columns
pd.set_option('display.max_columns', None)
```

Out[226]:

	Date	Post ID	Network	Post Type	Content Type	Profile	Sent by	
0	12/17/2022 5:08 pm	253788558082460_5441020446025886	Facebook	Post	Video	Stanbic IBTC	Damilare Oyekanmi	https://www.
1	2019-04-05 10:01:00	253788558082460_2001824979945467	Facebook	Post	Photo	Stanbic IBTC		https://www.
2	2020-02-06 21:00:00	253788558082460_2768996106561680	Facebook	Post	Photo	Stanbic IBTC		https://ww
3	2022-09-05 10:37:00	253788558082460_4808157902645480	Facebook	Post	Photo	Stanbic IBTC	Damilare Oyekanmi	https://ww
4	7/18/2021 10:00 am	253788558082460_3890983261029620	Facebook	Post	Photo	Stanbic IBTC		https://ww

In [227]:
```
df.tail()
```

Out[227]:

	Date	Post ID	Network	Post Type	Content Type	Profile	Sent by	
9798	2013-01-02 13:24:00	253788558082460_282819078512741	Facebook	Post	Photo	Stanbic IBTC		https://www.fa
9799	1/17/2013 5:08 pm	253788558082460_271983219596327	Facebook	Post	Photo	Stanbic IBTC		https://www.fa
9800	1/16/2013 4:58 pm	253788558082460_271477726313543	Facebook	Post	Photo	Stanbic IBTC		https://www.fa
9801	1/16/2013 4:51 pm	253788558082460_271474996313816	Facebook	Post	Photo	Stanbic IBTC		https://www.fa
9802	1/15/2013 4:25 pm	253788558082460_271028919691757	Facebook	Post	Link	Stanbic IBTC		https://www.face

In [229]:
```
df.columns
```

Out[229]:
```
Index(['Date', 'Post ID', 'Network', 'Post Type', 'Content Type', 'Profile', 'Sent b
y', 'Link', 'Post', 'Linked Content',
       ...
       'Video Removed from Playlists', 'Annotation Impressions', 'Annotation Clickabl
e Impressions', 'Annotation Closable Impressions', 'Annotation Closes', 'Card Impress
ions', 'Card Teaser Impressions', 'Card Teaser Clicks', 'Poll Votes', 'Tags'], dtype
='object', length=147)
```

In [228]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9803 entries, 0 to 9802
Columns: 147 entries, Date to Tags
dtypes: float64(136), object(11)
memory usage: 11.0+ MB
```

In [4]:
```
# List of chosen features
```

9802 1/15/2013 4:25 pm 253788558082460_271028919691757 Facebook Post Link Stanbic IBTC https://www.face

In [229]:
```
df.columns
```

Out[229]:
```
Index(['Date', 'Post ID', 'Network', 'Post Type', 'Content Type', 'Profile', 'Sent b
y', 'Link', 'Post', 'Linked Content',
       ...
       'Video Removed from Playlists', 'Annotation Impressions', 'Annotation Clickabl
e Impressions', 'Annotation Closable Impressions', 'Annotation Closes', 'Card Impress
ions', 'Card Teaser Impressions', 'Card Teaser Clicks', 'Poll Votes', 'Tags'], dtype
='object', length=147)
```

In [228]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9803 entries, 0 to 9802
Columns: 147 entries, Date to Tags
dtypes: float64(136), object(11)
memory usage: 11.0+ MB
```

In [4]:
```
# List of chosen features
chosen_features = [
    'Date','Content Type','Negative Feedback', 'Post', 'Impressions',
    'Engagements', 'Reactions', 'Comments', 'Shares', 'Click-Through Rate',
    'Non-fan Impressions','Reach','Viral Reach','Non-viral Reach','Engaged Fans',
    'Users Talking About This','Unique Post Clicks','Unique Reactions', 'Unique Comment
    'Fan Reach','Engaged Users','Viral Impressions','Non-viral Impressions',
    'Fan Organic Impressions','Post Clicks (All)','Love Reactions','Haha Reactions','Wo
```

In [5]:
```
# Select the chosen features from the DataFrame
df1 = df[chosen_features]
```

In [6]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9803 entries, 0 to 9802
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Date                      9803 non-null   object 
 1   Content Type              9803 non-null   object 
 2   Negative Feedback         8893 non-null   float64
 3   Post                      9553 non-null   object 
 4   Impressions               8893 non-null   float64
 5   Engagements               8893 non-null   float64
 6   Reactions                 8893 non-null   float64
 7   Comments                  8893 non-null   float64
 8   Shares                    8893 non-null   float64
 9   Click-Through Rate        8893 non-null   float64
 10  Non-fan Impressions       8893 non-null   float64
 11  Reach                     8893 non-null   float64
 12  Viral Reach               8893 non-null   float64
 13  Non-viral Reach           8893 non-null   float64
 14  Engaged Fans              8893 non-null   float64
 15  Users Talking About This  8893 non-null   float64
 16  Unique Post Clicks        8893 non-null   float64
 17  Unique Reactions          8893 non-null   float64
 18  Unique Comments           8893 non-null   float64
 19  Unique Shares             8893 non-null   float64
 20  Fan Reach                 8893 non-null   float64
 21  Engaged Users             8893 non-null   float64
 22  Viral Impressions         8893 non-null   float64
 23  Non-viral Impressions     8893 non-null   float64
 24  Fan Organic Impressions   8893 non-null   float64
 25  Post Clicks (All)         8893 non-null   float64
 26  Love Reactions            8893 non-null   float64
 27  Haha Reactions            8893 non-null   float64
 28  Wow Reactions             8893 non-null   float64
 29  Sad Reactions             8893 non-null   float64
 30  Angry Reactions           8893 non-null   float64
dtypes: float64(28), object(3)
memory usage: 2.3+ MB
```

In [7]:
```
#checking for missing values in the selected features
df1.isnull().sum()
```

Out[7]:
```
Date                          0
Content Type                  0
Negative Feedback           910
Post                        250
Impressions                 910
Engagements                 910
Reactions                   910
Comments                    910
Shares                      910
```

```
 28  Wow Reactions              8893 non-null   float64
 29  Sad Reactions              8893 non-null   float64
 30  Angry Reactions            8893 non-null   float64
dtypes: float64(28), object(3)
memory usage: 2.3+ MB
```

In [7]:
```
#checking for missing values in the selected features
df1.isnull().sum()
```

Out[7]:
```
Date                          0
Content Type                  0
Negative Feedback           910
Post                        250
Impressions                 910
Engagements                 910
Reactions                   910
Comments                    910
Shares                      910
Click-Through Rate          910
Non-fan Impressions         910
Reach                       910
Viral Reach                 910
Non-viral Reach             910
Engaged Fans                910
Users Talking About This    910
Unique Post Clicks          910
Unique Reactions            910
Unique Comments             910
Unique Shares               910
Fan Reach                   910
Engaged Users               910
Viral Impressions           910
Non-viral Impressions       910
Fan Organic Impressions     910
Post Clicks (All)           910
Love Reactions              910
Haha Reactions              910
Wow Reactions               910
Sad Reactions               910
Angry Reactions             910
dtype: int64
```

In [8]:
```
#checking the descriptive statistics of the data to ascertain the method to handle the
df1.describe()
```

Out[8]:

	Negative Feedback	Impressions	Engagements	Reactions	Comments	Shares	Click-Through Rate	Ir
count	8893.000000	8893.000000	8893.000000	8893.00000	8893.000000	8893.000000	8893.000000	8
mean	0.327561	5857.725177	231.683234	91.88699	20.538738	8.381311	0.001772	
std	0.775294	7295.418176	1107.892858	1014.84987	59.085734	19.115478	0.067465	3
min	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	
25%	0.000000	2019.000000	56.000000	23.00000	2.000000	0.000000	0.000000	
50%	0.000000	4082.000000	107.000000	39.00000	8.000000	2.000000	0.000000	
75%	0.000000	7376.000000	220.000000	76.00000	18.000000	10.000000	0.000000	
max	13.000000	207378.000000	72474.000000	70484.00000	1552.000000	588.000000	5.000000	206

In [9]:
```
#Replace missing values with the median for numerical columns
numerical_cols = df1.select_dtypes(include=[np.number])  # Select only numerical columr
df1[numerical_cols.columns] = numerical_cols.fillna(numerical_cols.median())
```

In [10]:
```
#checking data after treating missing values
df1.isnull().sum()
```

Out[10]:
```
Date                          0
Content Type                  0
Negative Feedback             0
Post                        250
Impressions                   0
Engagements                   0
Reactions                     0
Comments                      0
Shares                        0
```

```
numerical_cols = df1.select_dtypes(include=[np.number]) # Select only numerical column
```

In [10]:
```
df1[numerical_cols.columns] = numerical_cols.fillna(numerical_cols.median())
#checking data after treating missing values
df1.isnull().sum()
```

Out[10]:
```
Date                           0
Content Type                   0
Negative Feedback              0
Post                         250
Impressions                    0
Engagements                    0
Reactions                      0
Comments                       0
Shares                         0
Click-Through Rate             0
Non-fan Impressions            0
Reach                          0
Viral Reach                    0
Non-viral Reach                0
Engaged Fans                   0
Users Talking About This       0
Unique Post Clicks             0
Unique Reactions               0
Unique Comments                0
Unique Shares                  0
Fan Reach                      0
Engaged Users                  0
Viral Impressions              0
Non-viral Impressions          0
Fan Organic Impressions        0
Post Clicks (All)              0
Love Reactions                 0
Haha Reactions                 0
Wow Reactions                  0
Sad Reactions                  0
Angry Reactions                0
dtype: int64
```

In [11]:
```
# Drop rows with missing values
df1.dropna(inplace=True)
```

In [12]:
```
df1.isna().sum()
```

Out[12]:
```
Date                         0
Content Type                 0
Negative Feedback            0
Post                         0
Impressions                  0
Engagements                  0
Reactions                    0
Comments                     0
Shares                       0
Click-Through Rate           0
```

In [12]:
```
df1.isna().sum()
```

Out[12]:
```
Date                       0
Content Type               0
Negative Feedback          0
Post                       0
Impressions                0
Engagements                0
Reactions                  0
Comments                   0
Shares                     0
Click-Through Rate         0
Non-fan Impressions        0
Reach                      0
Viral Reach                0
Non-viral Reach            0
Engaged Fans               0
Users Talking About This   0
Unique Post Clicks         0
Unique Reactions           0
Unique Comments            0
Unique Shares              0
Fan Reach                  0
Engaged Users              0
Viral Impressions          0
Non-viral Impressions      0
Fan Organic Impressions    0
Post Clicks (All)          0
Love Reactions             0
Haha Reactions             0
Wow Reactions              0
Sad Reactions              0
Angry Reactions            0
dtype: int64
```

In [13]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9553 entries, 0 to 9797
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Date                      9553 non-null   object
 1   Content Type              9553 non-null   object
 2   Negative Feedback         9553 non-null   float64
 3   Post                      9553 non-null   object
 4   Impressions               9553 non-null   float64
```

In [13]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9553 entries, 0 to 9797
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Date                     9553 non-null   object
 1   Content Type             9553 non-null   object
 2   Negative Feedback        9553 non-null   float64
 3   Post                     9553 non-null   object
 4   Impressions              9553 non-null   float64
 5   Engagements              9553 non-null   float64
 6   Reactions                9553 non-null   float64
 7   Comments                 9553 non-null   float64
 8   Shares                   9553 non-null   float64
 9   Click-Through Rate       9553 non-null   float64
 10  Non-fan Impressions      9553 non-null   float64
 11  Reach                    9553 non-null   float64
 12  Viral Reach              9553 non-null   float64
 13  Non-viral Reach          9553 non-null   float64
 14  Engaged Fans             9553 non-null   float64
 15  Users Talking About This 9553 non-null   float64
 16  Unique Post Clicks       9553 non-null   float64
 17  Unique Reactions         9553 non-null   float64
 18  Unique Comments          9553 non-null   float64
 19  Unique Shares            9553 non-null   float64
 20  Fan Reach                9553 non-null   float64
 21  Engaged Users            9553 non-null   float64
 22  Viral Impressions        9553 non-null   float64
 23  Non-viral Impressions    9553 non-null   float64
 24  Fan Organic Impressions  9553 non-null   float64
 25  Post Clicks (All)        9553 non-null   float64
 26  Love Reactions           9553 non-null   float64
 27  Haha Reactions           9553 non-null   float64
 28  Wow Reactions            9553 non-null   float64
 29  Sad Reactions            9553 non-null   float64
 30  Angry Reactions          9553 non-null   float64
dtypes: float64(28), object(3)
memory usage: 2.3+ MB
```

In [77]:
```
# Convert 'Date' column to datetime format
df1['Date'] = pd.to_datetime(df1['Date'])
```

In [78]:
```
# Create new columns for year and time
df1['Year'] = df1['Date'].dt.year
df1['Time'] = df1['Date'].dt.time
```

In [79]:
```
# Function to extract the month
def extract_month(date):
    return date.strftime('%B')  # '%B' format returns the full month name
```

In [80]:
```
# Apply the extract_month function to create a new 'Month' column
df1['Month'] = df1['Date'].apply(extract_month)
```

In [81]:
```
# Define time of the day intervals
morning_start = pd.to_datetime('06:00:00').time()
afternoon_start = pd.to_datetime('12:00:00').time()
night_start = pd.to_datetime('18:00:00').time()

# Function to categorize time of day
def categorize_time_of_day(time):
    if time < morning_start:
        return 'Night'
    elif morning_start <= time < afternoon_start:
        return 'Morning'
    elif afternoon_start <= time < night_start:
```

In [81]:
```python
# Define time of the day intervals
morning_start = pd.to_datetime('06:00:00').time()
afternoon_start = pd.to_datetime('12:00:00').time()
night_start = pd.to_datetime('18:00:00').time()

# Function to categorize time of day
def categorize_time_of_day(time):
    if time < morning_start:
        return 'Night'
    elif morning_start <= time < afternoon_start:
        return 'Morning'
    elif afternoon_start <= time < night_start:
        return 'Afternoon'
    else:
        return 'Night'

# Apply the categorize_time_of_day function to create a new column
df1['Time_of_Day'] = df1['Time'].apply(categorize_time_of_day)

# Now, df1 contains a 'Time_of_Day' column with morning, afternoon, or night values
```

In [82]:
```python
# Extract the day of the week and create a new column for it
df1['Day_of_Week'] = df1['Date'].dt.day_name()
```

In [83]:
```python
df1.head(3)
```

Out[83]:

	Date	Content Type	Negative Feedback	Post	Impressions	Engagements	Reactions	Comments	Shares	C Thr
0	2022-12-17 17:08:00	Video	13.0	We celebrated recently with Novare, one of our...	207378.0	1024.0	179.0	59.0	2.0	
1	2019-04-05 10:01:00	Photo	10.0	N5k can get you started today. Call 01 280 126...	125784.0	5876.0	762.0	572.0	47.0	
2	2020-02-06 21:00:00	Photo	5.0	Still not sure whether to invest in the FGN Bo...	89699.0	4744.0	465.0	855.0	29.0	

In [27]:
```python
#descriptive statistics of the features to gain more insights
df1.describe().T
```

Out[27]:

	count	mean	std	min	25%	50%	75%	max
Negative Feedback	9553.0	0.297708	0.739621	0.0	0.0	0.0	0.0	13.0
Impressions	9553.0	5712.714959	6983.721195	0.0	2301.0	4082.0	6945.0	207378.0
Engagements	9553.0	220.907987	1067.973217	0.0	60.0	107.0	203.0	72474.0
Reactions	9553.0	87.415681	979.020761	0.0	24.0	39.0	70.0	70484.0
Comments	9553.0	19.474197	56.575201	0.0	3.0	8.0	17.0	1552.0

In [27]:

```python
#descriptive statistics of the features to gain more insights
df1.describe().T
```

Out[27]:

	count	mean	std	min	25%	50%	75%	max
Negative Feedback	9553.0	0.297708	0.739621	0.0	0.0	0.0	0.0	13.0
Impressions	9553.0	5712.714959	6983.721195	0.0	2301.0	4082.0	6945.0	207378.0
Engagements	9553.0	220.907987	1067.973217	0.0	60.0	107.0	203.0	72474.0
Reactions	9553.0	87.415681	979.020761	0.0	24.0	39.0	70.0	70484.0
Comments	9553.0	19.474197	56.575201	0.0	3.0	8.0	17.0	1552.0
Shares	9553.0	7.866325	18.467575	0.0	1.0	2.0	9.0	588.0
Click-Through Rate	9553.0	0.001641	0.065093	0.0	0.0	0.0	0.0	5.0
Non-fan Impressions	9553.0	899.106249	2954.153064	0.0	290.0	466.0	789.0	206231.0
Reach	9553.0	5096.316026	6047.396107	0.0	2149.0	3771.0	6181.0	207378.0
Viral Reach	9553.0	422.873338	1082.444702	0.0	7.0	83.0	385.0	24829.0
Non-viral Reach	9553.0	4672.990893	5506.360259	0.0	1967.0	3488.0	5746.0	213627.0
Engaged Fans	9553.0	111.186015	194.781823	0.0	32.0	61.0	118.0	4142.0
Users Talking About This	9553.0	67.083953	107.261190	0.0	22.0	39.0	72.0	1892.0
Unique Post Clicks	9553.0	79.054957	180.735924	0.0	17.0	38.0	76.0	4227.0
Unique Reactions	9553.0	56.285146	92.237316	0.0	18.0	33.0	60.0	1806.0
Unique Comments	9553.0	10.813985	32.311634	0.0	2.0	5.0	10.0	955.0
Unique Shares	9553.0	4.986287	10.122963	0.0	0.0	1.0	5.0	230.0
Fan Reach	9553.0	4343.859730	4789.075694	0.0	1789.0	3201.0	5359.0	76764.0
Engaged Users	9553.0	133.300429	233.301055	0.0	40.0	73.0	142.0	4514.0
Viral Impressions	9553.0	577.164660	1587.494128	0.0	8.0	88.0	453.0	35816.0
Non-viral Impressions	9553.0	5117.723752	6070.410378	0.0	2164.0	3784.0	6155.0	207373.0
Fan Organic Impressions	9553.0	4802.331624	5498.113106	0.0	1937.0	3483.0	5837.0	103287.0
Post Clicks (All)	9553.0	105.303884	258.882582	0.0	21.0	48.0	99.0	5989.0
Love Reactions	9553.0	2.119962	15.466986	0.0	0.0	0.0	1.0	471.0
Haha Reactions	9553.0	0.145713	1.547245	0.0	0.0	0.0	0.0	70.0
Wow Reactions	9553.0	0.118916	2.238990	0.0	0.0	0.0	0.0	163.0
Sad Reactions	9553.0	0.030776	0.217279	0.0	0.0	0.0	0.0	6.0
Angry Reactions	9553.0	0.140270	0.646311	0.0	0.0	0.0	0.0	21.0
Year	9553.0	2019.620852	2.280817	2013.0	2018.0	2020.0	2022.0	2023.0

```
Negative Feedback:
Mean: 0.298
Std: 0.740
Min: 0.0
Max: 13.0
Interpretation: The average negative feedback received per post is approximately 0.298,
with a wide range from 0 to a maximum of 13. Negative feedback may include reactions like
"angry" or "dislike."

Impressions:
Max: 72,474.0
Mean: 5712.71
Interpretation: The average engagement per post is about 220.91, including likes,
Std: 6983.72
comments, shares, etc. There is substantial variability, with some posts receiving no
Min: 0.0
engagement and others up to 72,474 engagements.
Max: 207,378.0
Reactions:
Interpretation: On average, posts were displayed to users around 5,712 times. However,
Mean: 87.42
there is significant variation, ranging from 0 to over 207,000 impressions.
Std: 979.02
Engagements:
Min: 0.0
Mean: 220.91
Max: 70,484.0
Std: 1067.97
Interpretation: Posts received an average of 87.42 reactions (e.g., likes, loves) per
Min: 0.0
post. However, there is a wide range, with some posts receiving no reactions and others
```

Max: 72,474.0
Interpretation: The average engagement per post is about 220.91, including likes, comments, shares, etc. There is substantial variability, with some posts receiving no engagement and others up to 72,474 engagements.

Impressions:
Mean: 5712.71
Std: 6983.70
Min: 0.0
Max: 207,378.0
Interpretation: On average, posts were displayed to users around 5,712 times. However, there is significant variation, ranging from 0 to over 207,000 impressions.

Reactions:
Mean: 87.42
Std: 979.02
Min: 0.0
Max: 70,408.0
Interpretation: Posts received an average of 87.42 reactions (e.g., likes, loves) per post. However, there is a wide range, with some posts receiving no reactions and others over 70,000.

Engagements:
Mean: 220.91
Std: 1067.97
Min: 0.0

Comments:
Mean: 19.47
Std: 56.58
Min: 0.0
Max: 1,552.0
Interpretation: On average, posts received approximately 19.47 comments. There is considerable variation, ranging from no comments to over 1,500.

Shares:
Mean: 7.87
Std: 18.47
Min: 0.0
Max: 588.0
Interpretation: Posts were shared an average of 7.87 times. However, the number of shares varies widely, with some posts having no shares and others up to 588 shares.

Click-Through Rate (CTR):
Mean: 0.0016
Std: 0.0651
Min: 0.0
Max: 5.0
Interpretation: The average CTR is very low at 0.0016, indicating that a small percentage of users clicked on links within the posts.

Non-fan Impressions:
Mean: 899.11
Std: 2954.15
Min: 0.0
Max: 206,231.0
Interpretation: Posts received an average of 899.11 impressions from non-fans. The number of non-fan impressions varies widely, with some posts having no non-fan impressions and others up to 206,231.

Reach:
Mean: 5096.32
Std: 6047.40
Min: 0.0
Max: 207,378.0
Interpretation: The average reach per post is about 5,096.32, indicating how many unique users saw the posts. There is significant variability, ranging from 0 to over 207,000 reach.

Viral Reach:
Mean: 422.87
Std: 1082.44
Min: 0.0
Max: 24,829.0
Interpretation: Posts reached an average of 422.88 users through viral means (e.g., sharing). The viral reach varies, with some posts having no viral reach and others up to 24,829.
Engaged Fans:
Mean: 111.19
Std: 194.78
Min: 0.0
Max: 4,142.0
Interpretation: On average, posts engaged 111.19 fans (followers of the page). However, there is variability, with some posts not engaging any fans and others engaging up to 4,142.

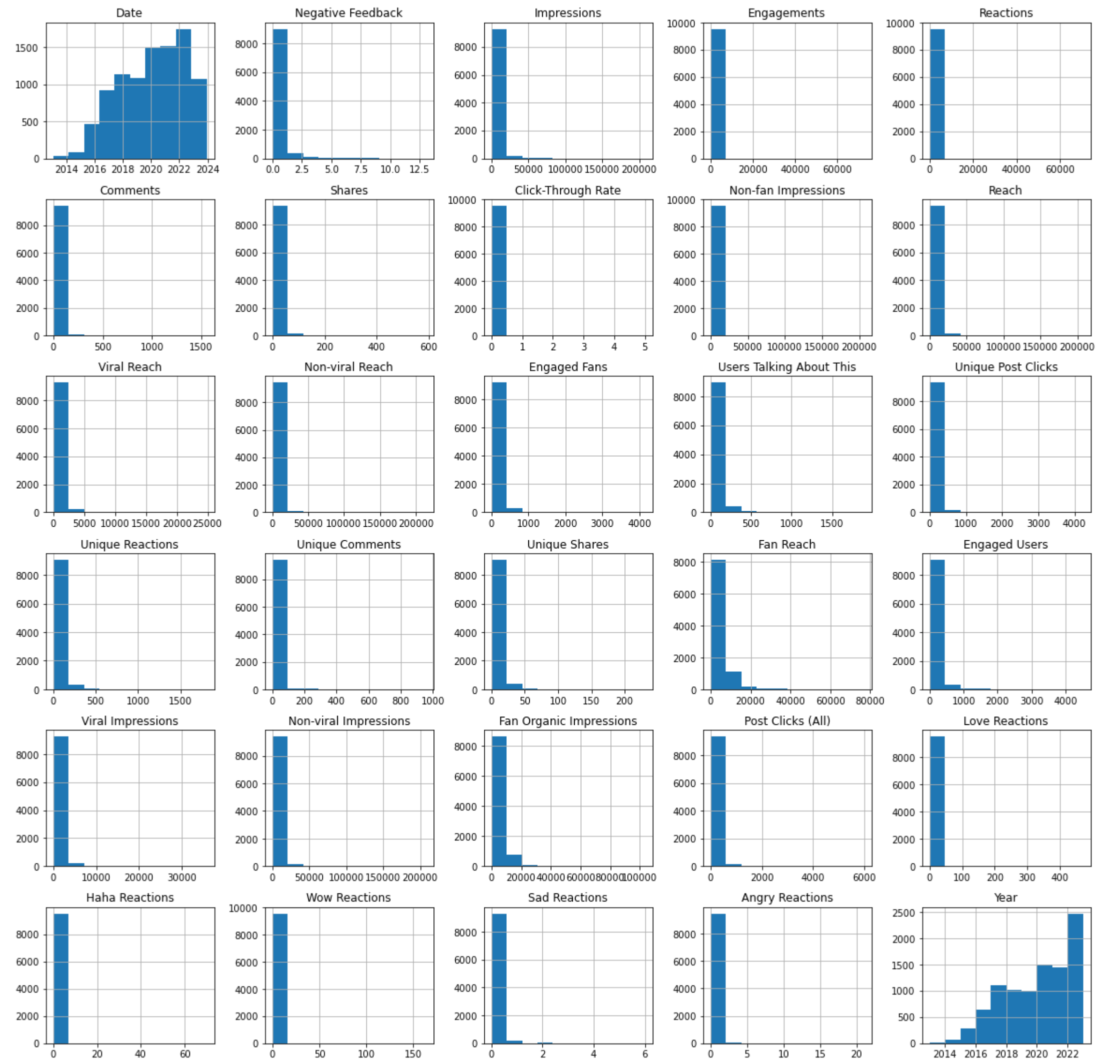
Non-viral Reach:
Mean: 4672.99
Std: 9506.36
Min: 0.0
Max: 213,627.0
Interpretation: Posts reached an average of 4,672.99 users through non-viral means. There is substantial variability, with some posts having no non-viral reach and others up to 213,627.

```
Max: 24,829.0
Interpretation: Posts reached an average of 422.88 users through viral means (e.g., sharing). The viral reach varies, with some posts having no viral reach and others up to 24,829.
Non-viral Reach:
Mean: 4672.99
Std: 9506.36
Min: 0.0
Max: 213,627.0
Interpretation: Posts reached an average of 4,672.99 users through non-viral means. There is substantial variability, with some posts having no non-viral reach and others up to 213,627.
Engaged Fans:
Mean: 111.19
Std: 194.78
Min: 0.0
Max: 4,142.0
Interpretation: On average, posts engaged 111.19 fans (followers of the page). However, there is variability, with some posts not engaging any fans and others engaging up to 4,142.

Users Talking About This:
Mean: 67.08
Std: 107.26
Min: 0.0
Max: 1,892.0
Interpretation: Users talked about posts an average of 67.08 times. There is variability, with some posts having no user discussions and others up to 1,892 discussions.

Unique Post Clicks:
Mean: 79.05
Std: 180.74
Min: 0.0
Max: 4,227.0
Interpretation: Posts received an average of 79.05 unique clicks on links within the posts. The number of unique clicks varies, with some posts having none and others up to 4,227.

Unique Reactions:
Mean: 56.29
Std: 92.24
Min: 0.0
Max: 1,806.0
Interpretation: Posts received an average of 56.29 unique reactions (e.g., unique likes or loves). There is variation, with some posts having no unique reactions and others up to 1,806.

Unique Comments:
Mean: 10.81
Std: 32.31
Min: 0.0
Max: 955.0
Interpretation: On average, posts received approximately 10.81 unique comments. However, there is variability, with some posts having no unique comments and others up to 955.

Unique Shares:
Mean: 4.99
Std: 10.12
Min: 0.0
Max: 230.0
Interpretation: Posts were shared an average of 4.99 times uniquely. There is variation, with some posts having no unique shares and others up to 230.

Fan Reach:
Mean: 4343.86
Std: 4789.08
Min: 0.0
Max: 76,764.0
Interpretation: Posts reached an average of 4,343.86 fans. There is variability, with some posts not reaching any fans and others reaching up to 76,764.
Viral Impressions:
Mean: 577.16
Std: 1587.49
Min: 0.0
Max: 35,816.0
Interpretation: Posts received an average of 577.16 viral impressions (views due to sharing). The number of viral impressions varies, with some posts having none and others up to 35,816.
Engaged Users:
Mean: 133.30
Std: 235.90
Min: 0.0
Max: 4,514.0
Interpretation: On average, 133.30 users engaged with posts (e.g., liked, commented, shared). There is variability, with some posts not engaging any users and others engaging up to 4,514.
```

In [28]:
```python
#chart showing the distribution of the data
df1.hist()
plt.gcf().set_size_inches(20,20)
```

```
Interpretation: Posts reached an average of 4,343.86 fans. There is variability, with some posts not reaching any fans and others reaching up to 76,764.
Viral Impressions:
Mean: 577.16
Std: 1587.49
Min: 0.0
Max: 35381.0
Interpretation: Posts received an average of 577.16 viral impressions (views due to sharing). The number of viral impressions varies, with some posts having none and others up to 35,381.
Engaged Users:
Mean: 133.30
Std: 235.38
Min: 0.0
Max: 4514.0
Interpretation: On average, 133.30 users engaged with posts (e.g., liked, commented, shared). There is variability, with some posts not engaging any users and others engaging up to 4,514.
```

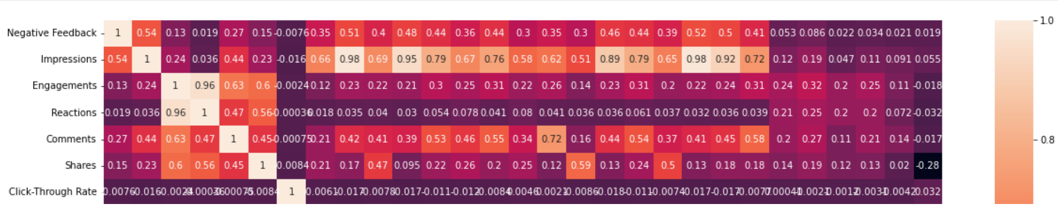
In [28]:
```python
#chart showing the distribution of the data
df1.hist()
plt.gcf().set_size_inches(20,20)
plt.show()
```



Based on the distribution from the histogram, we can observe that the engagements on the post are generally skewed towards the lower end of the spectrum. This means that, in a significant portion of the posts, the number of engagements, which includes likes, comments, shares, and other interactions, tends to be relatively low.

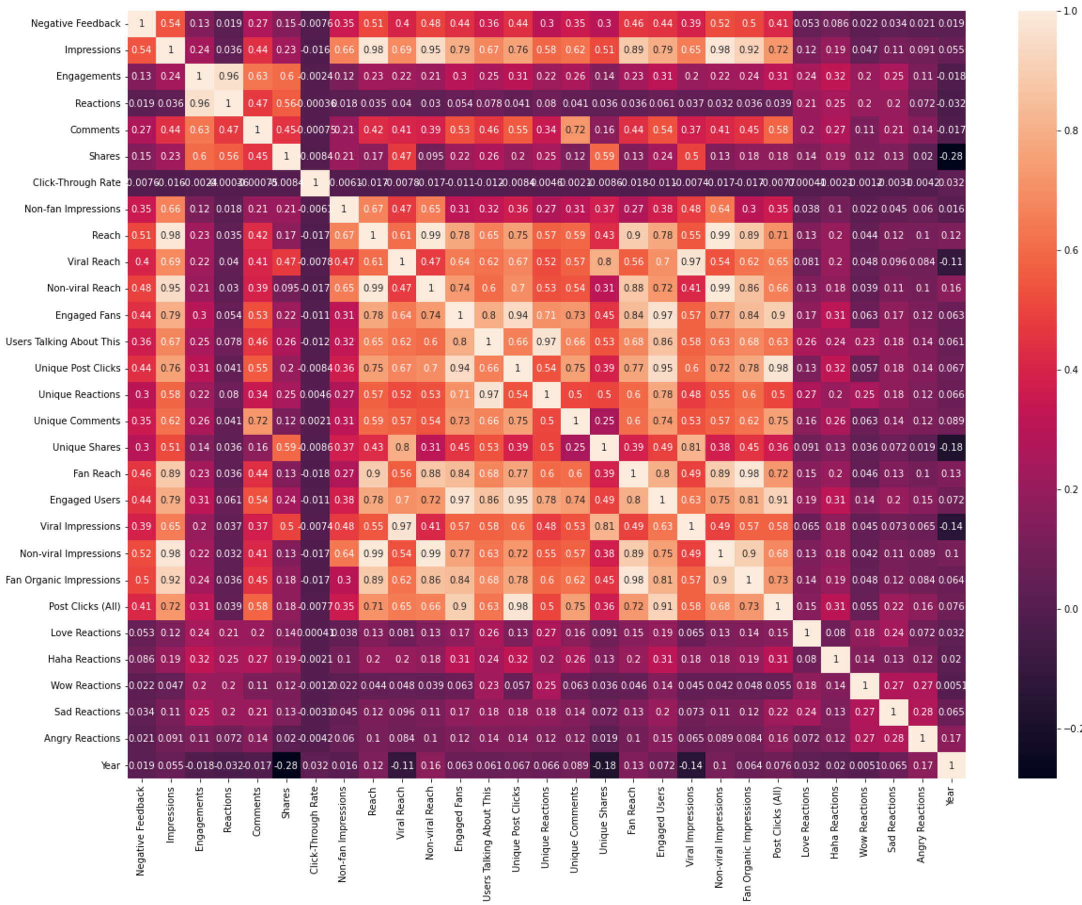
Specifically, the histogram shows that a substantial number of posts receive a low number of engagements, as indicated by the peaks and clustering of data points on the left side of the histogram. These posts may have received minimal user interaction, such as a few likes or a couple of comments.

In [29]:
```python
#heatmap showing the correlation of the variables
plt.figure(figsize = (20,15))
sns.heatmap(df1.corr(), annot = True)
set_option ('display.width', 1000)
```



indicated by the peaks and clustering of data points on the left side of the histogram. These posts may have received minimal user interaction, such as a few likes or a couple of comments.

In [29]:
```
#heatmap showing the correlation of the variables
plt.figure(figsize = (20,15))
sns.heatmap(df1.corr(), annot = True)
set_option ('display.width', 1000)
```



This heatmap provides valuable insights into the level of correlation between different variables within our dataset. It's evident that there is a range of correlation strengths among these variables, and some correlations are notably stronger than others. Let's delve into the key findings:

Negative Feedback and Impressions:

Negative feedback exhibits a moderately strong positive correlation with impressions, with a coefficient of 0.54. This suggests that posts receiving more negative feedback also tend to have a higher number of impressions. It's worth exploring the reasons behind this relationship. Impressions and Other Metrics:

Impressions show strong positive correlations with several other metrics, including fan organic impressions, non-viral impressions, non-viral reach, and overall reach, all of which have correlation coefficients above 0.9. This indicates that posts with higher impressions tend to also generate greater organic and non-viral reach. Engagements and Reactions:

Engagements demonstrate the highest positive correlation with reactions, with a remarkably high coefficient of 0.96. This suggests that when users engage with posts (e.g., through likes, comments, and shares), they are highly likely to react to the content. Comments and Engagement:

Comments display a significant positive correlation with both unique comments and overall engagement, both of which have correlation coefficients exceeding 0.6. This implies that posts with more comments tend to generate higher overall engagement levels. Shares and Other Metrics:

Shares exhibit positive correlations with viral impressions, unique shares, reactions, and engagements, all of which have correlation coefficients above 0.5. This implies that posts that are shared more frequently also tend to receive more reactions and higher overall engagement. Reach and Related Metrics:

Reach demonstrates a strong positive correlation with several metrics, including post clicks (all), fan organic impressions, non-viral impressions, engaged users, fan reach, unique post clicks, engaged fans, non-viral reach, and impressions, all of which have correlation coefficients above 0.7. This suggests that posts with greater reach tend to lead to higher engagement and interaction with the audience. Viral Reach and Viral Impression:

Viral reach displays a strong positive correlation with viral impressions, indicating that posts with higher viral reach also tend to generate more viral impressions. Engaged Fans and Engagement Metrics:

Engaged fans have positive correlations with post clicks (all), fan organic impressions, fan reach, engaged users, unique post clicks (all), and users talking about this, all of which have coefficients above 0.7. This suggests that posts with a higher number of engaged fans tend to result in increased user interactions and engagement. Unique Shares and Viral Metrics:

Unique shares exhibit a strong positive correlation with both viral impressions and viral reach, with correlation

Comments display a significant positive correlation with both unique comments and overall engagement, both
greater reach tend to lead to higher engagement and interaction with the audience. Viral Reach and Viral
of which have correlation coefficients exceeding 0.6. This implies that posts with more comments tend to
Impression:
generate higher overall engagement levels. Shares and Other Metrics:
Viral reach displays a strong positive correlation with viral impressions, indicating that posts with higher viral
Shares exhibit positive correlations with viral impressions, unique shares, reactions, and engagements, all of
reach also tend to generate more viral impressions. Engaged Fans and Engagement Metrics:
which have correlation coefficients above 0.5. This implies that posts that are shared more frequently also tend
Engaged fans have positive correlations with post clicks (all), fan organic impressions, fan reach, engaged
to receive more reactions and higher overall engagement. Reach and Related Metrics:
users, unique post clicks (all), and users talking about this, all of which have coefficients above 0.7. This
Reach demonstrates a strong positive correlation with several metrics, including post clicks (all), fan organic
suggests that posts with a higher number of engaged fans tend to result in increased user interactions and
impressions, non-viral impressions, engaged users, fan reach, unique post clicks, engaged fans, non-viral
engagement. Unique Shares and Viral Metrics:
reach, and impressions, all of which have correlation coefficients above 0.7. This suggests that posts with
Unique shares exhibit a strong positive correlation with both viral impressions and viral reach, with correlation coefficients exceeding 0.8. This implies that posts with a higher number of unique shares tend to go viral more frequently.

These findings provide valuable insights into the relationships between different metrics within your dataset. Understanding these correlations can help identify key drivers of engagement and optimize your social media content strategy accordingly.

In [ ]:

```

```

In [30]:

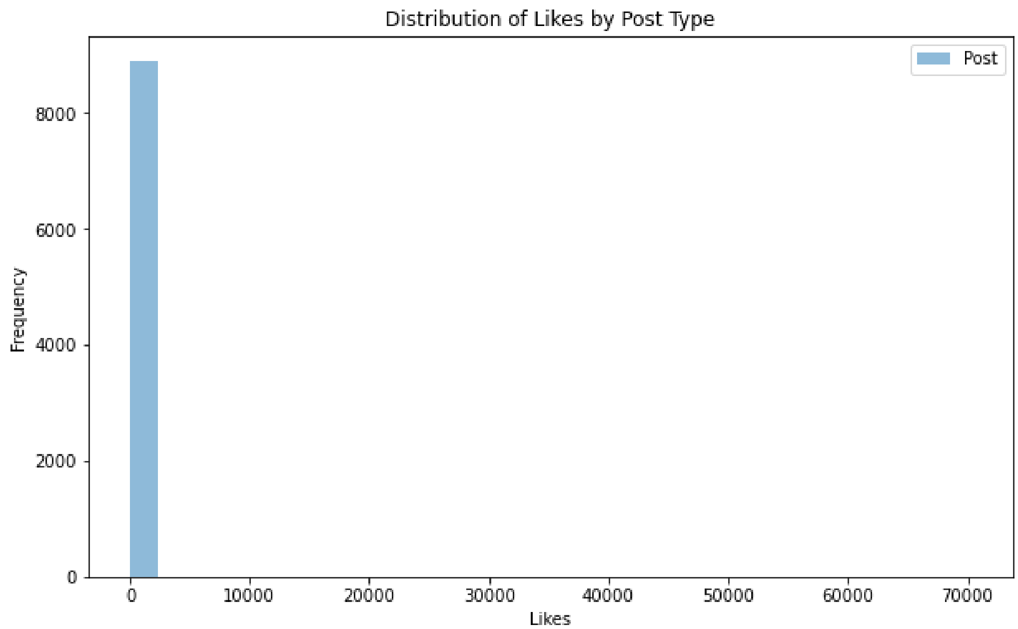
```
# Explore the distribution of different types of reactions by post type
reaction_columns = ['Likes', 'Love Reactions', 'Haha Reactions', 'Wow Reactions', 'Sad
```

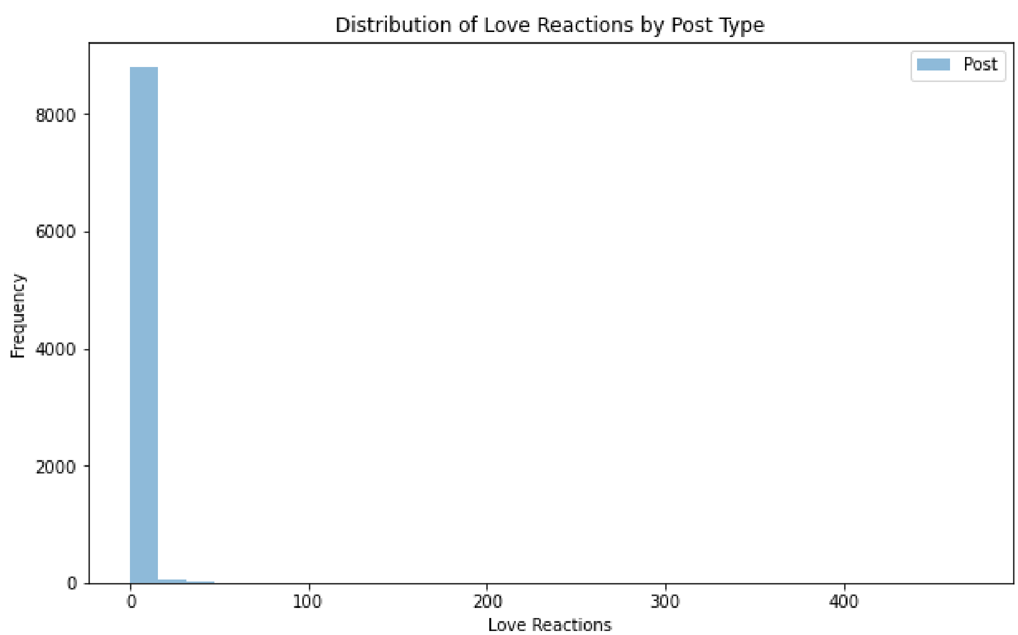
In [31]:

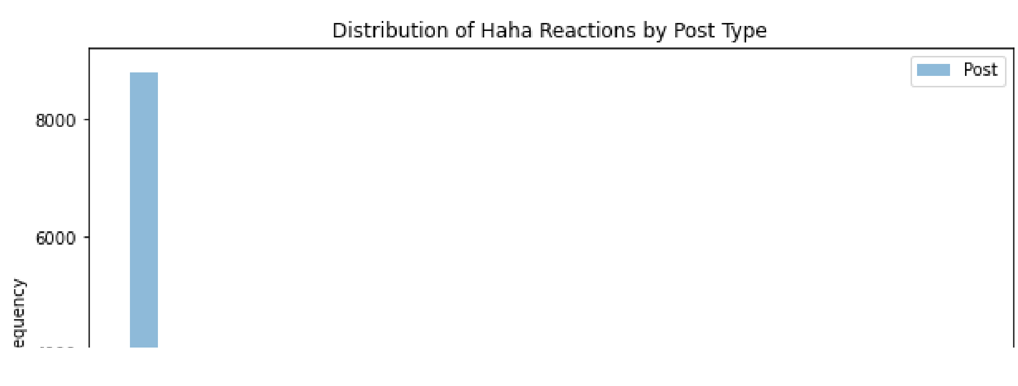
```
for reaction in reaction_columns:
    plt.figure(figsize=(10, 6))
    for post_type in df['Post Type'].unique():
        data = df[df['Post Type'] == post_type]
        plt.hist(data[reaction], bins=30, alpha=0.5, label=post_type)

    plt.title(f'Distribution of {reaction} by Post Type')
    plt.xlabel(reaction)
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

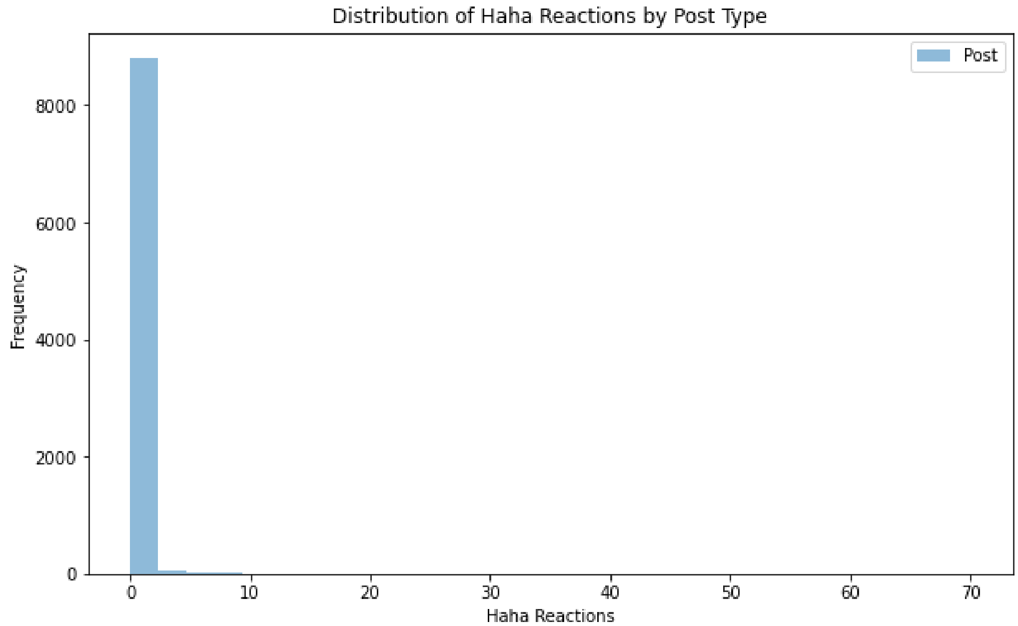
In [31]:

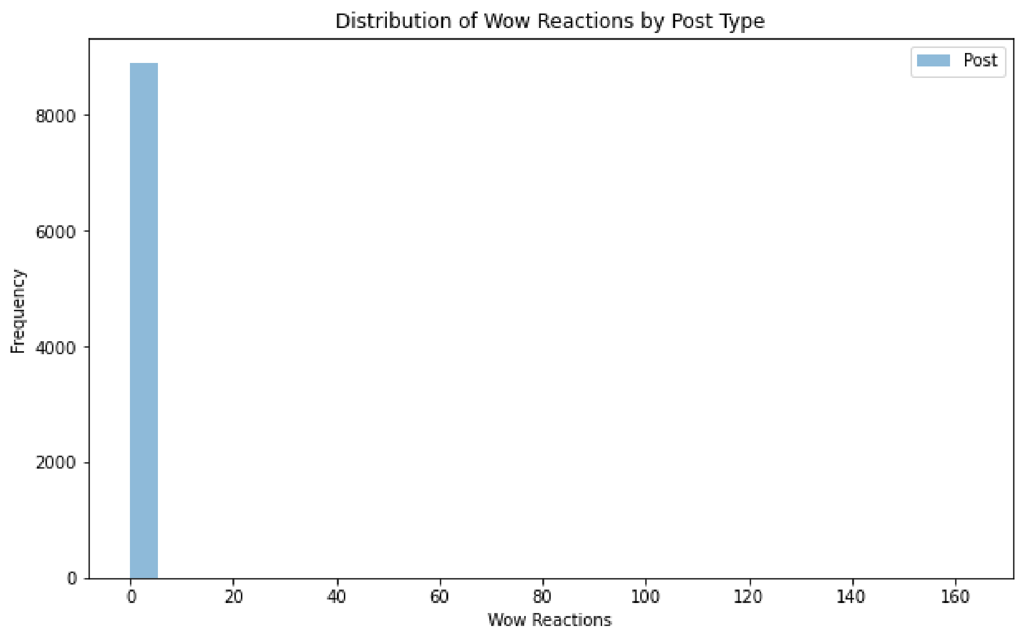
```
for reaction in reaction_columns:
    plt.figure(figsize=(10, 6))
    for post_type in df['Post Type'].unique():
        data = df[df['Post Type'] == post_type]
        plt.hist(data[reaction], bins=30, alpha=0.5, label=post_type)

    plt.title(f'Distribution of {reaction} by Post Type')
    plt.xlabel(reaction)
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

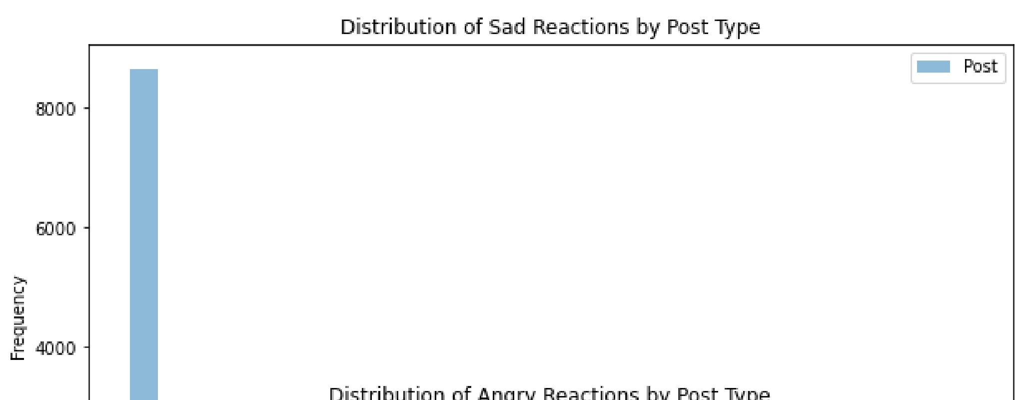


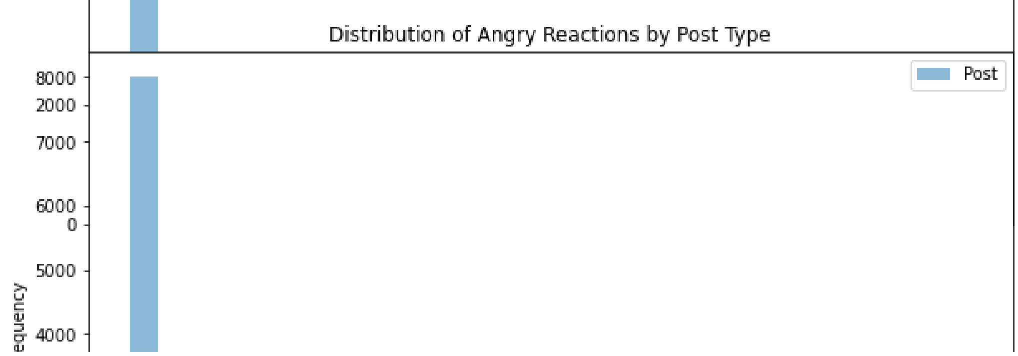


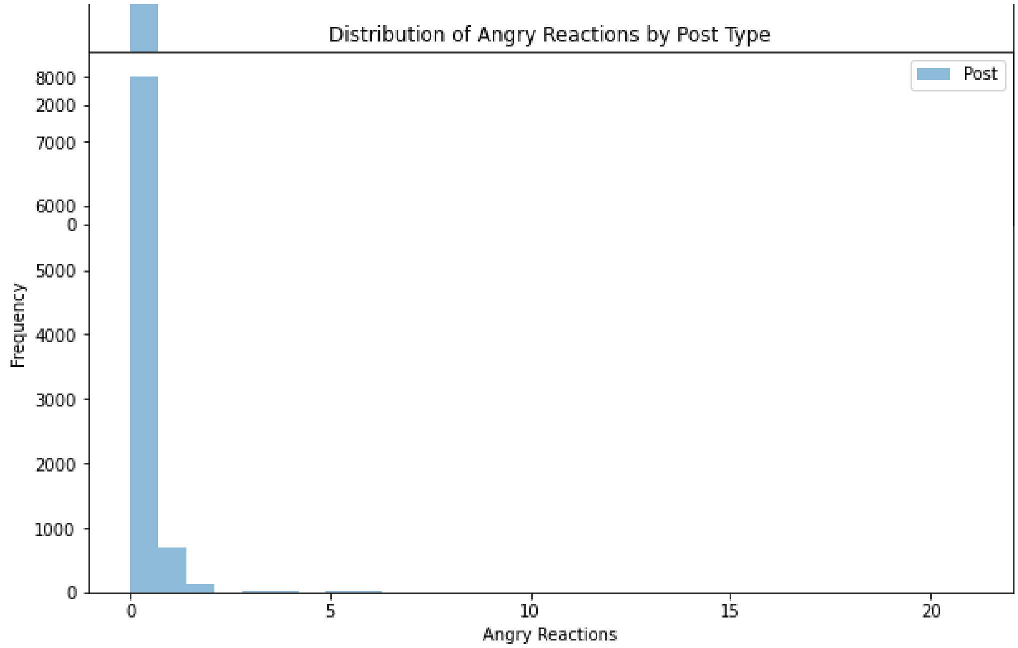



Distribution of Haha Reactions by Post Type
Post
Frequency
Haha Reactions


Distribution of Wow Reactions by Post Type
Post
Frequency
Wow Reactions


Distribution of Sad Reactions by Post Type
Post
Frequency


Distribution of Angry Reactions by Post Type
Post



From these charts, it's evident that certain types of reactions—specifically, "Love," "Sad," "Angry," and "Haha" reactions—generated some of the highest frequencies among the various types of user interactions with your Facebook posts. These reactions are essential indicators of how your audience is emotionally engaging with your content.

To gain deeper insights, it would be beneficial to further explore the specific posts that elicited these types of reactions. By analyzing these posts in greater detail, you can uncover patterns or content strategies that resonate particularly well with your audience and drive these emotional reactions.

Understanding why certain posts generate higher frequencies of these specific reactions can inform your content strategy. For example, you might discover that heartwarming or emotionally resonant stories tend to evoke "Love" reactions, while controversial topics might lead to more "Angry" reactions. Identifying these patterns can help you tailor your future content to maximize the desired emotional responses from your audience and enhance overall engagement.

In summary, these charts serve as a starting point for investigating the posts that generated these specific reactions, allowing you to refine your content strategy and create more impactful and engaging Facebook posts.

In [32]:
```
#exploring the top 5 post with the highest reactions
post = df.groupby('Post')[['Likes']].sum().sort_values(by=['Likes'],ascending= False)
```

In [33]:
```
post.head(5)
```

Out[33]:

Post	Likes
Avoid carrying papers. Open a Stanbic IBTC account here https://instantaccount.stanbicibtc.com:9663/index.html \nIt's simple and easy	70244.0
You can open an account with your eyes closed. Almost. Click to open - https://instantaccount.stanbicibtc.com:9663/index.html	57786.0
Repost #MyDreamsCanBe	18709.0
Life isn't easy. Banking should be. #NewInternetBanking will have you hooked. Visit stanbicibtcbank/internetbanking to get started.	15811.0
With a unique gift of Mutual funds from Stanbic IBTC, you can give your loved ones more reasons to celebrate this Easter season. So go ahead and express your love through our Mutual Funds.\n \nLog on to http://bit.ly/1sRqxwX and invest in their lives today or contact us on +234 1280 5595, 0700 MUTUALFUNDS (0700 6888 2538 637), or email: mutualfunds@stanbicibtc.com	13944.0

In [34]:
```
#exploring the top 5 posts with love reactions
love = df.groupby('Post')[['Love Reactions']].sum().sort_values(by=['Love Reactions'],a
```

In [35]:
```
love.head(5)
```

Out[35]:

Post	Love Reactions
Repost #MyDreamsCanBe	859.0

Repost #MyDreamsCanBe	18709.0
Life isn,Äôt easy. Banking should be. #NewInternetBanking will have you hooked. Visit stanbicibtcbank/internetbanking to get started.	15811.0
With a unique gift of Mutual funds from Stanbic IBTC, you can give your loved ones more reasons to celebrate this Easter season. So go ahead and express your love through our Mutual Funds.\n \nLog on to http://bit.ly/1sRqxwX and invest in their lives today or contact us on +234 1280 5595, 0700 MUTUALFUNDS (0700 6888 2538 637), or email: mutualfunds@stanbicibtc.com	13944.0

In [34]:
```
#exploring the top 5 posts with love reactions
love = df.groupby('Post')[['Love Reactions']].sum().sort_values(by=['Love Reactions'],a
```

In [35]:
```
love.head(5)
```

Out[35]:

Post	Love Reactions
Repost #MyDreamsCanBe	859.0
Always make sure you drink enough water to stay hydrated as you #StayHome during this period. #StaySafe	471.0
On this #WorldHealthDay, we celebrate the Nurses, Doctors and all healthcare workers working tirelessly during this COVID-19 Pandemic. Thank you for your selfless service.	394.0
Reward4Saving Live Draw - September\n\nToday, seven people will be rewarded with N1million each and additional Seventy people will walk away with N100,000 each for saving!\nYou too can be one of them in the next Reward4Saving live draw. All you need to do is deposit N10,000 into a new or existing savings account or @easewallet account to qualify.\n\n#Reward4Saving\n#ITCANBE	389.0
Remember to maintain the best health practices; regularly wash your hands with soap and water. \nAs they say, Health is Wealth. #StaySafe	373.0

In [36]:
```
#exploring the top 5 posts with sad reactions
sad = df.groupby('Post')[['Sad Reactions']].sum().sort_values(by=['Sad Reactions'],asce
```

In [37]:
```
sad.head(5)
```

Out[37]:

Post	Sad Reactions
Repost #MyDreamsCanBe	6.0
Reward4Saving Live Draw - September\n\nToday, seven people will be rewarded with N1million each and additional Seventy people will walk away with N100,000 each for saving!\nYou too can be one of them in the next Reward4Saving live draw. All you need to do is deposit N10,000 into a new or existing savings account or @easewallet account to qualify.\n\n#Reward4Saving\n#ITCANBE	6.0
Reward4Saving Live Draw - October 2022\n\n70 people will get N100,000 richer today, at the #Reward4Saving October 2022 live draw. \nWill you be one? Let's find out.\nTo qualify, deposit N10,000 into a new or existing savings account or @easewallet for 30 days.\n\n#ITCANBE	5.0
You can open an account with your eyes closed. Almost. Click to open - https://instantaccount.stanbicibtc.com:9663/index.html	4.0
üôåif you got it right! #Trivia	3.0

In [38]:
```
#exploring the top 5 posts with angry reactions
angry = df.groupby('Post')[['Angry Reactions']].sum().sort_values(by=['Angry Reactions'
```

In [39]:
```
angry.head(5)
```

Out[39]:

Post	Angry Reactions
Repost #MyDreamsCanBe	21.0
Every human life is a precious gift to humanity. \nLet us love and respect one another. \nWe,Äôre one Africa; let,Äôs put aside all differences and stand together with love. \nWe #SayNoToXenophobia	20.0
With the Stanbic IBTC Dollar Fund, be rest assured that no matter what your financial needs are, our	

In [39]:
```python
angry.head(5)
```

Out[39]:

Post	Angry Reactions
Repost #MyDreamsCanBe	21.0
Every human life is a precious gift to humanity. \nLet us love and respect one another. \nWe,Äôre one Africa; let,Äôs put aside all differences and stand together with love. \nWe #SayNoToXenophobia	20.0
With the Stanbic IBTC Dollar Fund, be rest assured that no matter what your financial needs are, our investment product can meet those needs. \nVisit https://bit.ly/StanbicIBTCDollarFund or send a mail to assetmanagement@stanbicibtc.com to get started. \n\n#WealthWednesday\n#ITCANBE\n#GoForIt	14.0
Reward4Saving Live Draw - January 2023\n\nLet's make money rain again as 70 people will get N100,000 richer today, at the #Reward4Saving January 2023 live draw. \nWill you be one? Let's find out.\nTo qualify, deposit N10,000 into a new or existing savings account or @easewallet for 30 days.\n\n#ITCANBE	14.0
We sincerely apologise for any inconvenience you may be experiencing now in trying to use our banking channels. The upgrade is in progress and an update will be shared once full services are restored. In the meantime, you can reach out to us by calling 0700 909 909 909 or by sending an email to CustomerCareNigeria@stanbicibtc.com for any enquiries.	10.0

In [40]:
```python
#exploring the top 5 posts with haha reactions
haha = df.groupby('Post')[['Haha Reactions']].sum().sort_values(by=['Haha Reactions'],a
```

In [41]:
```python
haha.head(5)
```

Out[41]:

Post	Haha Reactions
#HappenToLife with a Stanbic IBTC Pension and let us help you protect your future. For more information, please call 01 271 6000 or email pensionsolution@stanbicibtc.com #Pension #RetireWell #Future	70.0
What song tops your favourite playlist right now?	63.0
If you were born in June, ,Äòcut soap for us oh,Äô! □üòÑ \n #ITCANBE	63.0
Reminder: Don,Äôt forget to deposit your old Naira notes at any of our branches. You have five days to go! What are you waiting for?	62.0
You can open an account with your eyes closed. Almost. Click to open - https://instantaccount.stanbicibtc.com:9663/index.html	29.0

Negative Feedback Analysis

In [42]:
```python
#exploring the top 5 posts with negative feedback
negative = df1.groupby('Post')[['Negative Feedback']].sum().sort_values(by=['Negative F
```

In [43]:
```python
negative.head(5)
```

Out[43]:

Post	Negative Feedback
Do you know the name and what this local snack is made from?	13.0
We celebrated recently with Novare, one of our top clients on the 6th anniversary of their Novare Lekki mall opening. \n#TrustedPartner\n#ITCANBE	13.0
Repost #MyDreamsCanBe	12.0
Find the missing letters\nH_ _e_1_ _e\n\nHint ,Äì It,Äôs a movie of a boy who was forgotten at home.	11.0
N5k can get you started today. Call 01 280 1266 or email assetmanagement@stanbicibtc.com.	10.0

In [ ]:

In [44]:
```python
#content type analysis
(df1['Content Type'].value_counts())
```

Out[44]:
```
Photo    8118
Video     942
Text      325
Link      168
Name: Content Type, dtype: int64
```

We celebrated recently with Novare, one of our top clients on the 6th anniversary of their Novare Lekki mall opening. \n#TrustedPartner\n#ITCANBE 13.0

Repost #MyDreamsCanBe 12.0

Find the missing letters\nH_ _e_ _l_ _e\n\nHint ,Äì It,Äôs a movie of a boy who was forgotten at home. 11.0

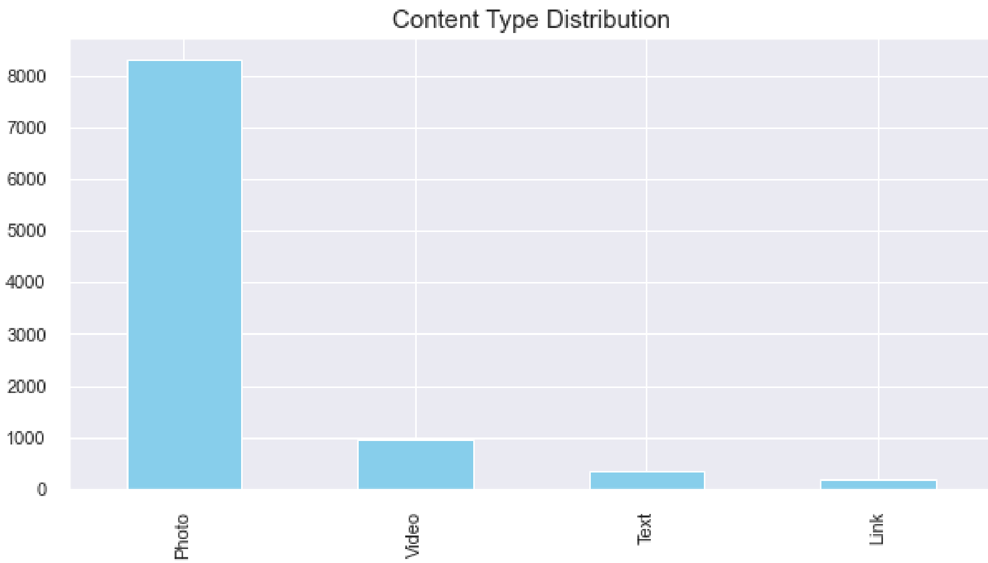
N5k can get you started today. Call 01 280 1266 or email assetmanagement@stanbicibtc.com. 10.0

In [ ]:

In [44]:
```
#content type analysis
(df1['Content Type'].value_counts())
```

Out[44]:
```
Photo    8118
Video     942
Text      325
Link      168
Name: Content Type, dtype: int64
```

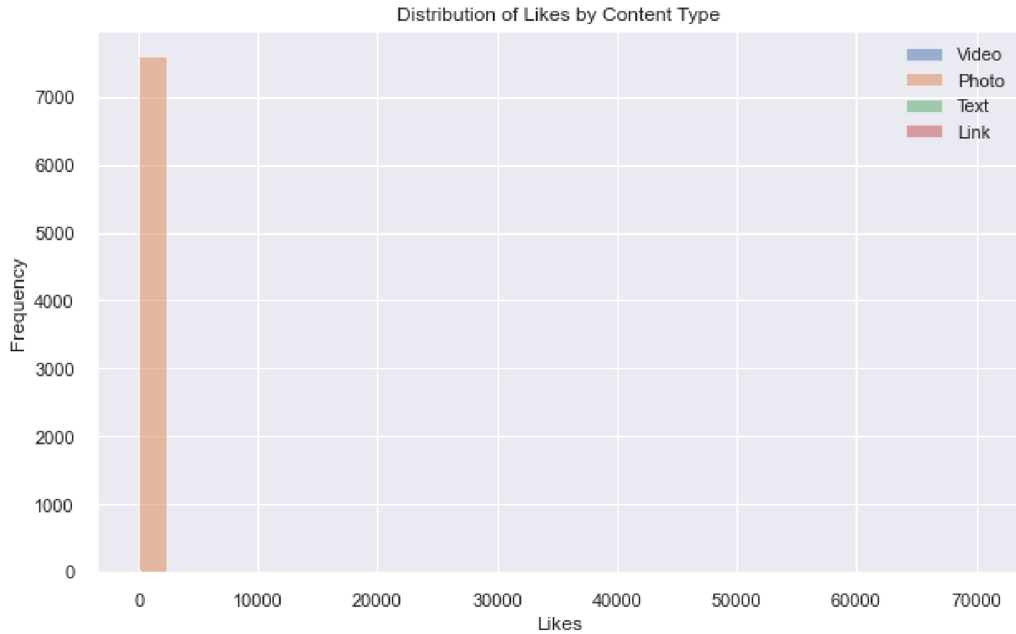
In [45]:
```
plt.style.use('seaborn')
sns.set(style="darkgrid")
plt.figure(figsize = (10, 5))
plt.title('Content Type Distribution', fontsize = 15)
df['Content Type'].value_counts()[:5].plot(kind='bar', color='skyblue')
plt.show()
```

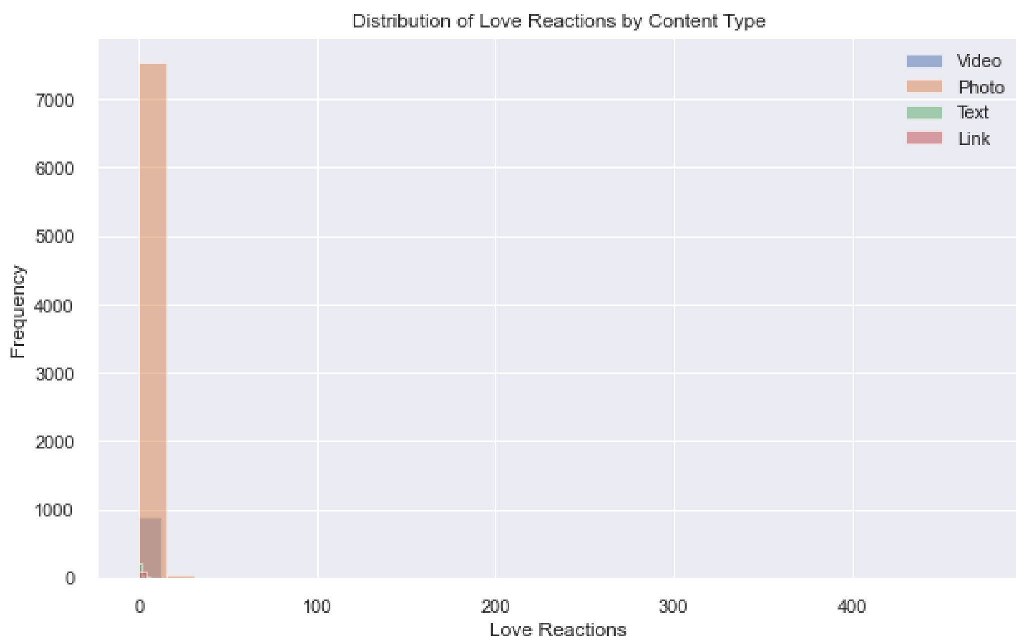


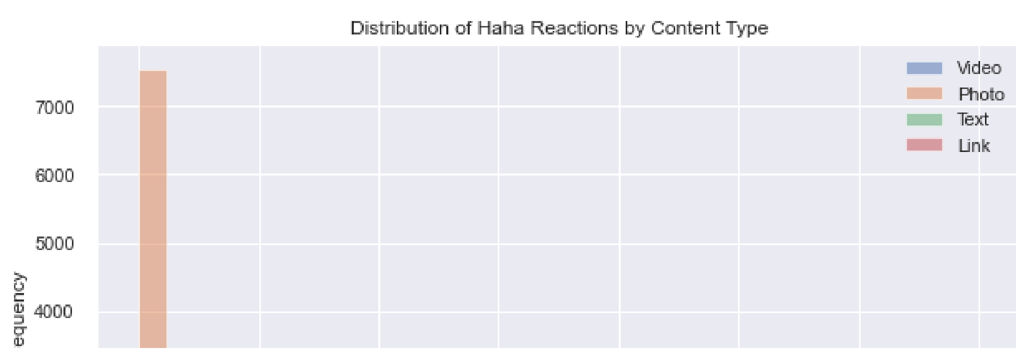
In [46]:
```
# Explore the distribution of different types of reactions by content type
for reaction in reaction_columns:
    plt.figure(figsize=(10, 6))
    for content_type in df['Content Type'].unique():
        data = df[df['Content Type'] == content_type]
        plt.hist(data[reaction], bins=30, alpha=0.5, label=content_type)

    plt.title(f'Distribution of {reaction} by Content Type')
    plt.xlabel(reaction)
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

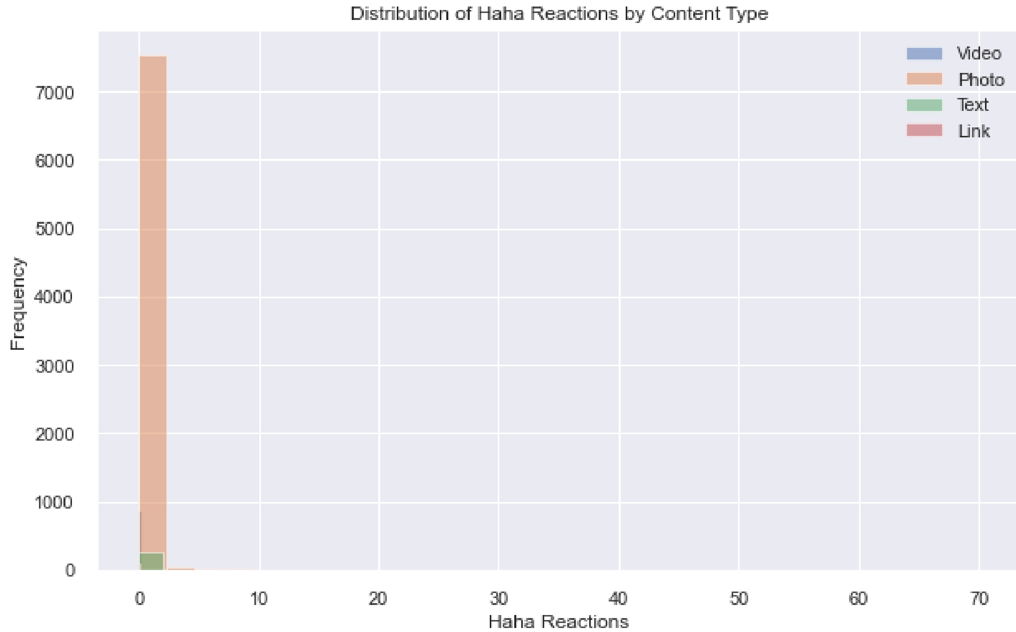
In [46]:

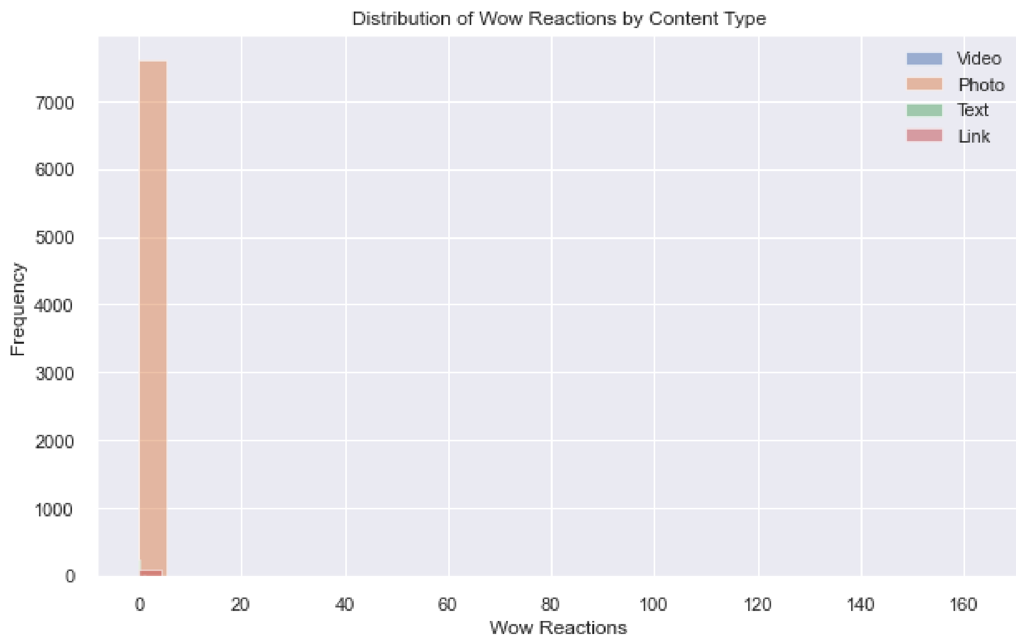
```python
# Explore the distribution of different types of reactions by content type
for reaction in reaction_columns:
    plt.figure(figsize=(10, 6))
    for content_type in df['Content Type'].unique():
        data = df[df['Content Type'] == content_type]
        plt.hist(data[reaction], bins=30, alpha=0.5, label=content_type)

    plt.title(f'Distribution of {reaction} by Content Type')
    plt.xlabel(reaction)
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

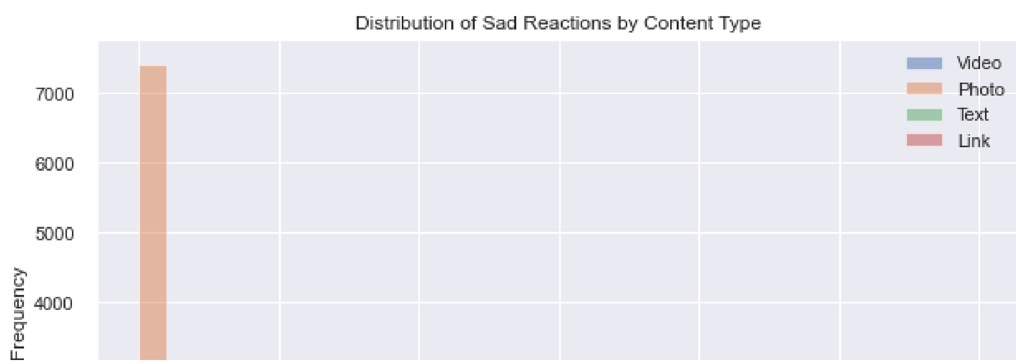


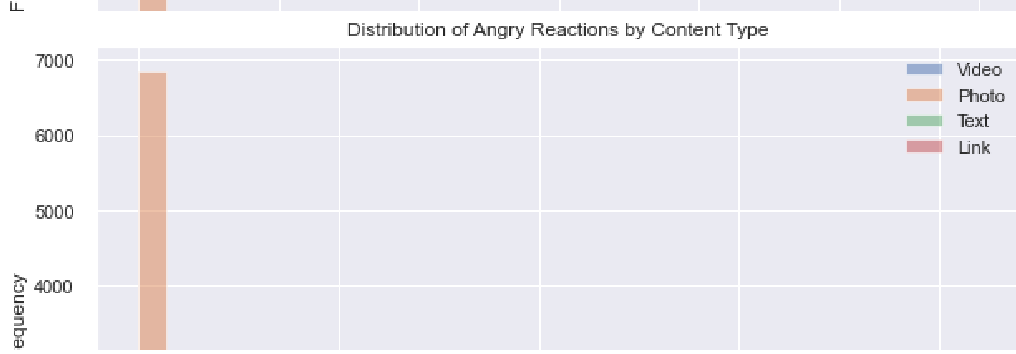


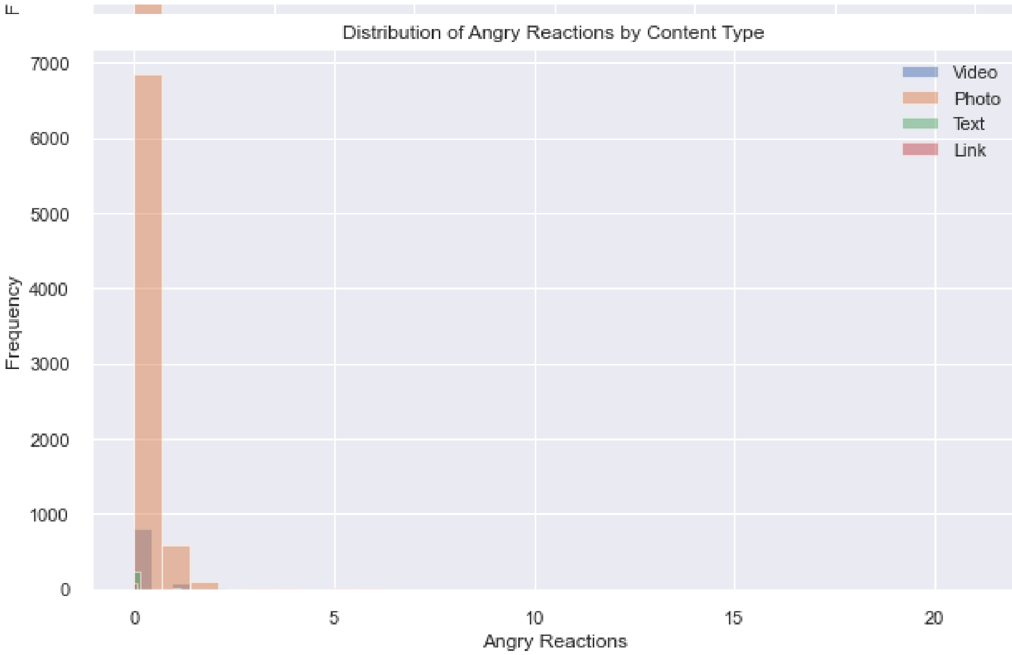



Distribution of Haha Reactions by Content Type
Video
Photo
Text
Link
Frequency
0
1000
2000
3000
4000
5000
6000
7000
0
10
20
30
40
50
60
70
Haha Reactions


Distribution of Wow Reactions by Content Type
Video
Photo
Text
Link
Frequency
0
1000
2000
3000
4000
5000
6000
7000
0
20
40
60
80
100
120
140
160
Wow Reactions


Distribution of Sad Reactions by Content Type
Video
Photo
Text
Link
Frequency
4000
5000
6000
7000

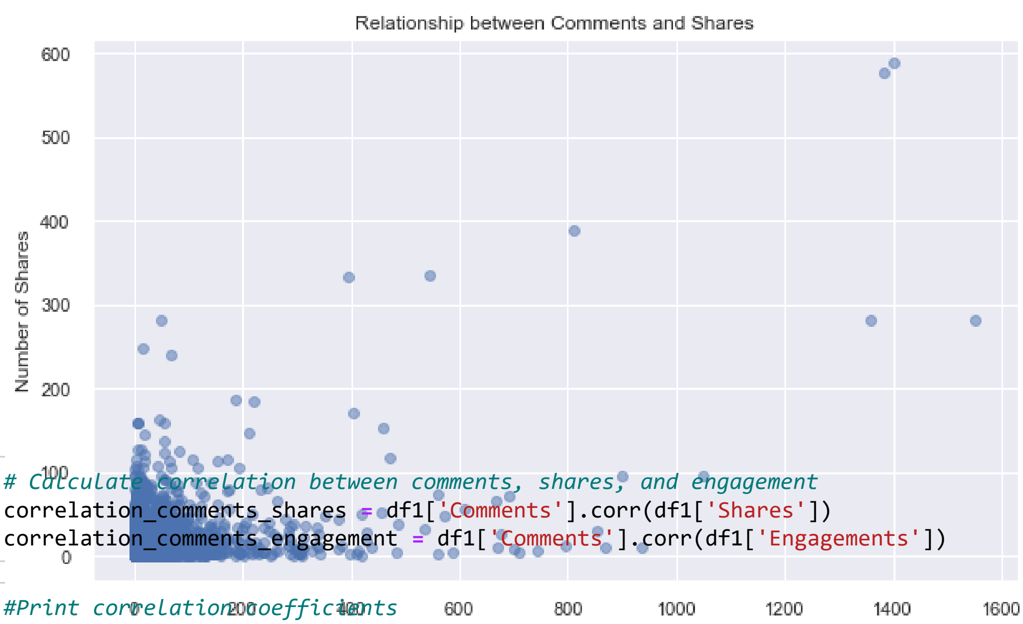

Distribution of Angry Reactions by Content Type
Video
Photo
Text
Link
equency
4000
5000
6000
7000



The analysis of content types reveals a notable trend: Facebook posts featuring photos consistently attract the highest levels of user engagement, including various types of reactions. This observation aligns with the composition of the dataset, where a significant proportion of the posts consist of photos. In light of these findings, it becomes apparent that incorporating visual elements, especially photos, into your content strategy can be a potent tool for enhancing user engagement on Facebook. This approach not only leverages the dataset's existing composition but also aligns with the broader trend of visual content's effectiveness in captivating and connecting with online audiences.

In [47]:

```
# Investigate the relationship between comments, shares, and engagement
plt.figure(figsize=(10, 6))

# Scatter plot of comments vs. shares
plt.scatter(df1['Comments'], df1['Shares'], alpha=0.5)
plt.title('Relationship between Comments and Shares')
plt.xlabel('Number of Comments')
plt.ylabel('Number of Shares')
plt.grid(True)
```



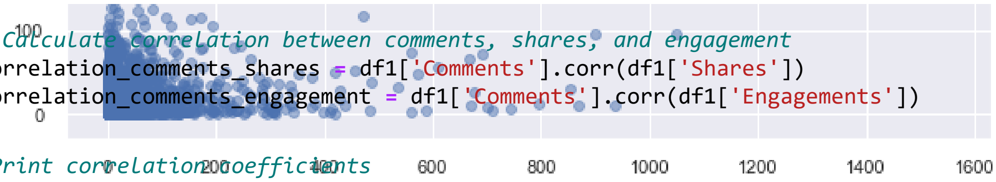
In [48]:

```
# Calculate correlation between comments, shares, and engagement
correlation_comments_shares = df1['Comments'].corr(df1['Shares'])
correlation_comments_engagement = df1['Comments'].corr(df1['Engagements'])
```

In [49]:

```
#Print correlation coefficients
print(f'Correlation between Comments and Shares: {correlation_comments_shares}')
print(f'Correlation between Comments and Engagement: {correlation_comments_engagement}'
```

```
Correlation between Comments and Shares: 0.4513040229288868
Correlation between Comments and Engagement: 0.6301812432633006
```



In [48]:
```
# Calculate correlation between comments, shares, and engagement
correlation_comments_shares = df1['Comments'].corr(df1['Shares'])
correlation_comments_engagement = df1['Comments'].corr(df1['Engagements'])
```

In [49]:
```
#Print correlation coefficients
print(f'Correlation between Comments and Shares: {correlation_comments_shares}')
print(f'Correlation between Comments and Engagement: {correlation_comments_engagement}'
```

```
Correlation between Comments and Shares: 0.4513040229288868
Correlation between Comments and Engagement: 0.6301812432633006
```

The correlation coefficients between comments and shares, as well as comments and engagement, provide valuable insights into the relationships among these variables:

Correlation between Comments and Shares (0.451):

A positive correlation coefficient of approximately 0.451 suggests a moderate positive relationship between the number of comments on Facebook posts and the number of shares those posts receive. This means that, in general, when a post attracts a higher number of comments, it is more likely to also have a higher number of shares. These two engagement metrics are positively associated. A potential interpretation is that engaging and discussion-provoking content tends to be shared more frequently. Users may comment on a post to express their opinions or engage in discussions, and this, in turn, may lead to more shares as others share the content to participate in the conversation. Correlation between Comments and Engagement (0.630):

A stronger positive correlation coefficient of approximately 0.630 indicates a relatively strong positive relationship between the number of comments on Facebook posts and the overall engagement those posts receive. This means that posts with a higher number of comments tend to have higher overall engagement levels, which includes reactions, shares, and other forms of interaction. The interpretation here is that comments play a significant role in driving overall engagement. When users actively comment on a post, it tends to attract more reactions, shares, and other forms of engagement. This could be because comments often signify a deeper level of engagement and interaction with the content. In summary, both correlations highlight the importance of user comments in fostering engagement on Facebook posts. Posts that encourage meaningful discussions and receive a higher number of comments are likely to also see increased shares and overall engagement. These insights can inform your content strategy, emphasizing the value of encouraging user participation and conversations within your Facebook posts to boost their reach and impact.

Time analysis

In [50]:
```
df1.head(3)
```

Out[50]:

	Date	Content Type	Negative Feedback	Post	Impressions	Engagements	Reactions	Comments	Shares	C Thr
0	2022-12-17 17:08:00	Video	13.0	We celebrated recently with Novare, one of our...	207378.0	1024.0	179.0	59.0	2.0	
1	2019-04-05 10:01:00	Photo	10.0	N5k can get you started today. Call 07280 126...	125784.0	5876.0	762.0	572.0	47.0	
2	2020-02-06 21:00:00	Photo	5.0	Still not sure whether to invest in the FGN Bo...	89699.0	4744.0	465.0	855.0	29.0	

In [84]:
```
# Extract the hour of the day and create a new column for it
df1['Hour_of_Day'] = df1['Date'].dt.hour
```

In [85]:
```
df1.head(2)
```

Out[85]:

	Date	Content Type	Negative Feedback	Post	Impressions	Engagements	Reactions	Comments	Shares	C Thr
				We celebrated						

	Date	Content Type	Negative Feedback	Post	Impressions	Engagements	Reactions	Comments	Shares
1	2019-04-05 10:01:00	Photo	10.0	... today. Call 01 280 126...	125784.0	5876.0	762.0	572.0	47.0
2	2020-02-06 21:00:00	Photo	5.0	Still not sure whether to invest in the FGN Bo...	89699.0	4744.0	465.0	855.0	29.0

In [84]:
```
# Extract the hour of the day and create a new column for it
df1['Hour_of_Day'] = df1['Date'].dt.hour
```

In [85]:
```
df1.head(2)
```

Out[85]:

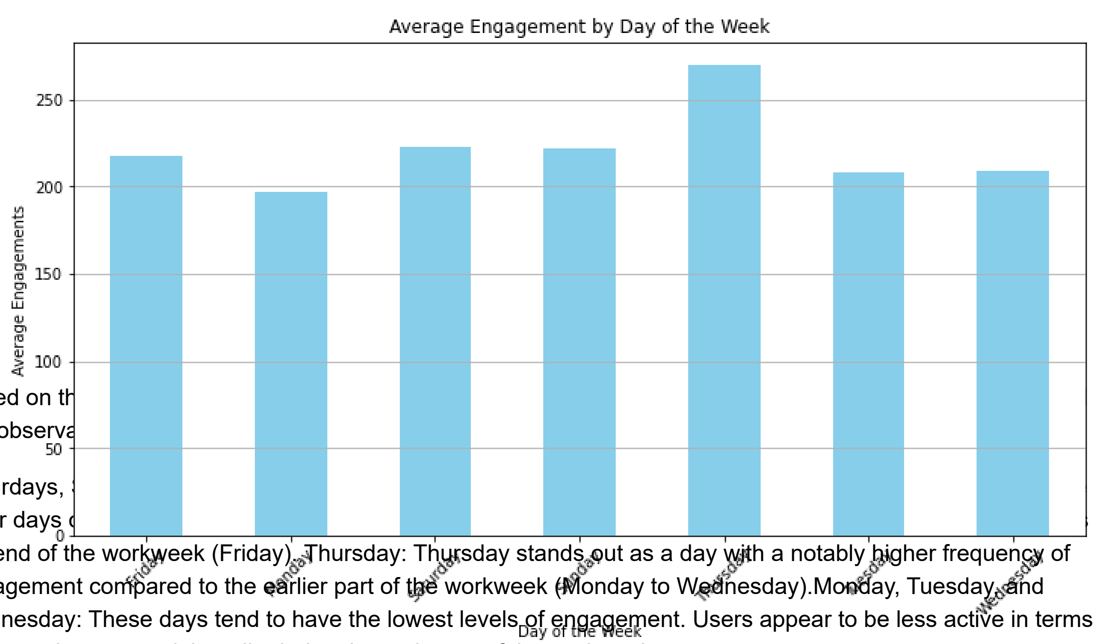
	Date	Content Type	Negative Feedback	Post	Impressions	Engagements	Reactions	Comments	Shares	C Thr
0	2022-12-17 17:08:00	Video	13.0	We celebrated recently with Novare, one of our...	207378.0	1024.0	179.0	59.0	2.0	
1	2019-04-05 10:01:00	Photo	10.0	N5k can get you started today. Call 01 280 126...	125784.0	5876.0	762.0	572.0	47.0	

In [86]:
```
# Explore engagement by day of the week
engagement_by_day = df1.groupby('Day_of_Week')['Engagements'].mean()
# Explore engagement by hour of the day
engagement_by_hour = df1.groupby('Hour_of_Day')['Engagements'].mean()
```

In [87]:
```
# Explore reach by day of the week
reach_by_day = df1.groupby('Day_of_Week')['Reach'].mean()
```

In [88]:
```
# Explore reach by hour of the day
reach_by_hour = df1.groupby('Hour_of_Day')['Reach'].mean()
```

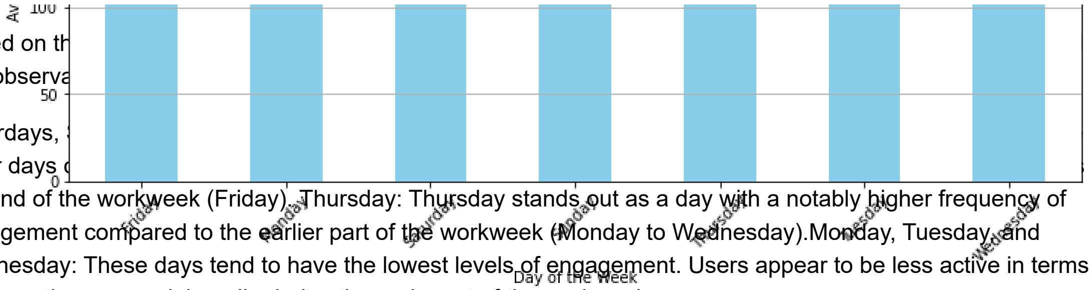
In [89]:
```
# Plot engagement by day of the week
plt.figure(figsize=(12, 6))
engagement_by_day.plot(kind='bar', color='skyblue')
plt.title('Average Engagement by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Average Engagements')
plt.xticks(rotation=45)
plt.grid(axis='y')
```



Based on th... key observa...

Saturdays, ... other days ... the end of the workweek (Friday). Thursday: Thursday stands out as a day with a notably higher frequency of engagement compared to the earlier part of the workweek (Monday to Wednesday). Monday, Tuesday, and Wednesday: These days tend to have the lowest levels of engagement. Users appear to be less active in terms of interactions on social media during the early part of the workweek.

These observations provide valuable insights into the patterns of user engagement on different days of the

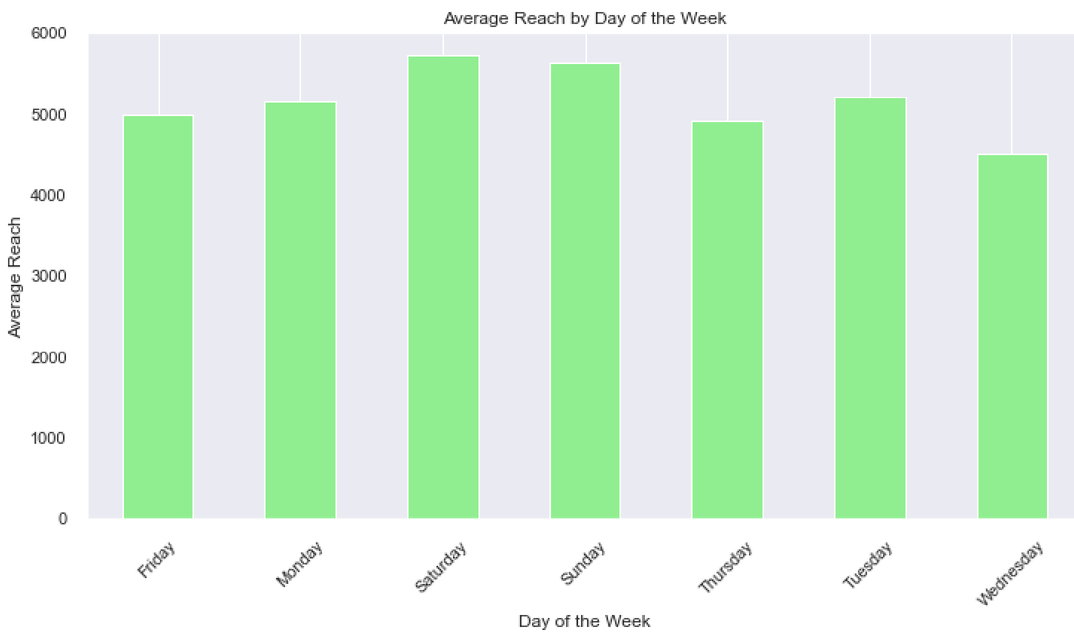


Based on th
key observa

Saturdays,
other days
the end of the workweek (Friday). Thursday: Thursday stands out as a day with a notably higher frequency of engagement compared to the earlier part of the workweek (Monday to Wednesday).Monday, Tuesday, and Wednesday: These days tend to have the lowest levels of engagement. Users appear to be less active in terms of interactions on social media during the early part of the workweek.

These observations provide valuable insights into the patterns of user engagement on different days of the week. It suggests that when planning your content strategy, you may want to consider scheduling more engaging or impactful posts during weekends and Fridays when user activity is higher. Conversely, you might adjust your strategy for Mondays, Tuesdays, and Wednesdays when engagement tends to be lower.

In [58]:
```python
# Plot reach by day of the week
plt.figure(figsize=(12, 6))
reach_by_day.plot(kind='bar', color='lightgreen')
plt.title('Average Reach by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Average Reach')
plt.xticks(rotation=45)
plt.grid(axis='y')
```



In the analysis of reach across different days of the week, it's evident that each day consistently garners a respectable level of reach. However, Saturdays and Sundays distinctly emerge as the standout performers in this regard.

Consistent Reach: The data reveals that your content maintains a commendable level of reach throughout the week, indicating that your audience remains engaged and connected with your posts on a daily basis.

Weekend Peaks: Saturdays and Sundays, in particular, demonstrate a noteworthy spike in reach. These days consistently achieve the highest reach metrics, signifying that your content tends to resonate exceptionally well with your audience over the weekend.

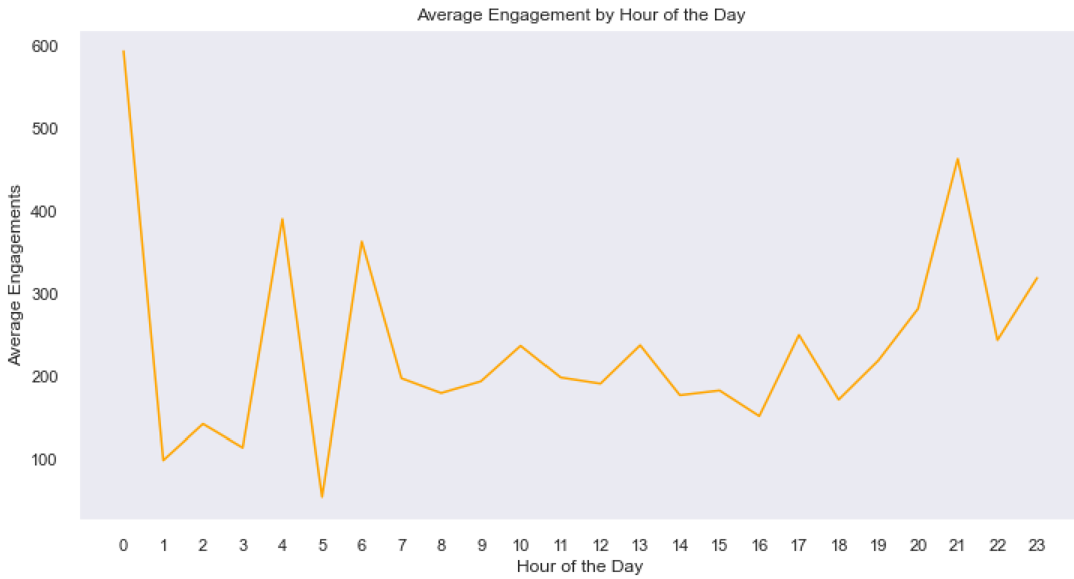
This observation underscores the significance of weekends in your content strategy. Leveraging the weekends, when reach is at its peak, may provide opportunities to maximize the impact of your posts, potentially reaching a broader and more receptive audience. Understanding these patterns empowers you to tailor your posting schedule to align with when your content is most likely to achieve optimal reach and engagement.

In [59]:
```python
# Plot engagement by hour of the day
plt.figure(figsize=(12, 6))
engagement_by_hour.plot(kind='line', color='orange')
plt.title('Average Engagement by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Engagements')
plt.xticks(range(24))
plt.grid()
```

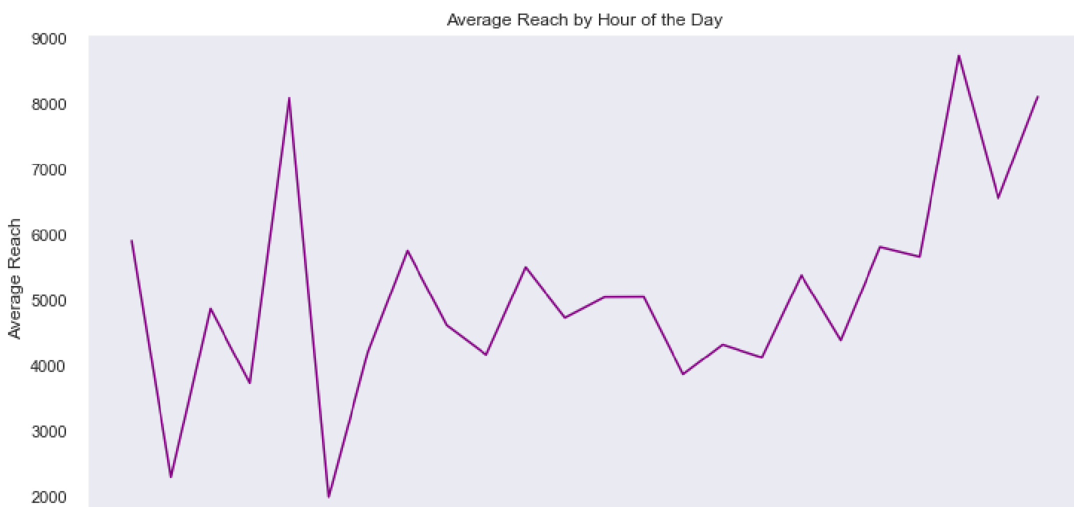
consistently achieve the highest reach metrics, signifying that your content tends to resonate exceptionally well with your audience over the weekend.

This observation underscores the significance of weekends in your content strategy. Leveraging the weekends, when reach is at its peak, may provide opportunities to maximize the impact of your posts, potentially reaching a broader and more receptive audience. Understanding these patterns empowers you to tailor your posting schedule to align with when your content is most likely to achieve optimal reach and engagement.

In [59]:
```
# Plot enggagement by hour of the day
plt.figure(figsize=(12, 6))
engagement_by_hour.plot(kind='line', color='orange')
plt.title('Average Engagement by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Engagements')
plt.xticks(range(24))
plt.grid()
```



In [60]:
```
# Plot reach by hour of the day
plt.figure(figsize=(12, 6))
reach_by_hour.plot(kind='line', color='purple')
plt.title('Average Reach by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Reach')
plt.xticks(range(24))
plt.grid()

plt.show()
```



In [62]:
```
# Explore reach by time of day
reach_by_time = df1.groupby('Time_of_Day')['Reach'].mean()
```

In [61]:
```
# Explore engagement by time of day
engagement_by_time = df1.groupby('Time_of_Day')['Engagements'].mean()
```

In [63]:
```
# Plot enggagement by time of day
plt.figure(figsize=(12, 6))
engagement_by_time.plot(kind='line', color='skyblue')
plt.title('Average Engagement by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Average Engagements')
plt.xticks(rotation=45)
plt.grid()
```

In [62]:

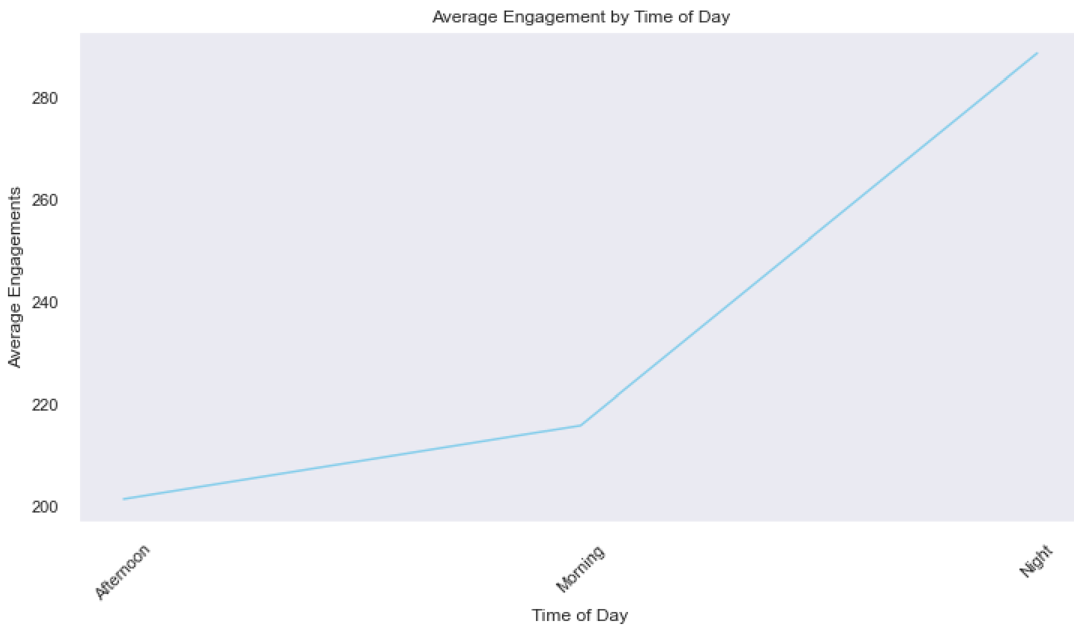
```python
# Explore reach by time of day
reach_by_time = df1.groupby('Time_of_Day')['Reach'].mean()
```

In [61]:

```python
# Explore engagement by time of day
engagement_by_time = df1.groupby('Time_of_Day')['Engagements'].mean()
```

In [63]:

```python
# Plot engagement by time of day
plt.figure(figsize=(12, 6))
engagement_by_time.plot(kind='line', color='skyblue')
plt.title('Average Engagement by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Average Engagements')
plt.xticks(rotation=45)
plt.grid()
```



The chart depicting engagement levels throughout the day offers valuable insights into the timing patterns of user interactions with contents. Here's a refined interpretation:

Morning to Night Trend: The data illustrates a clear upward trajectory in engagement starting from the morning hours and reaching its zenith during the nighttime. This suggests that users tend to become progressively more engaged with your posts as the day unfolds, with the peak occurring during the evening and nighttime periods.

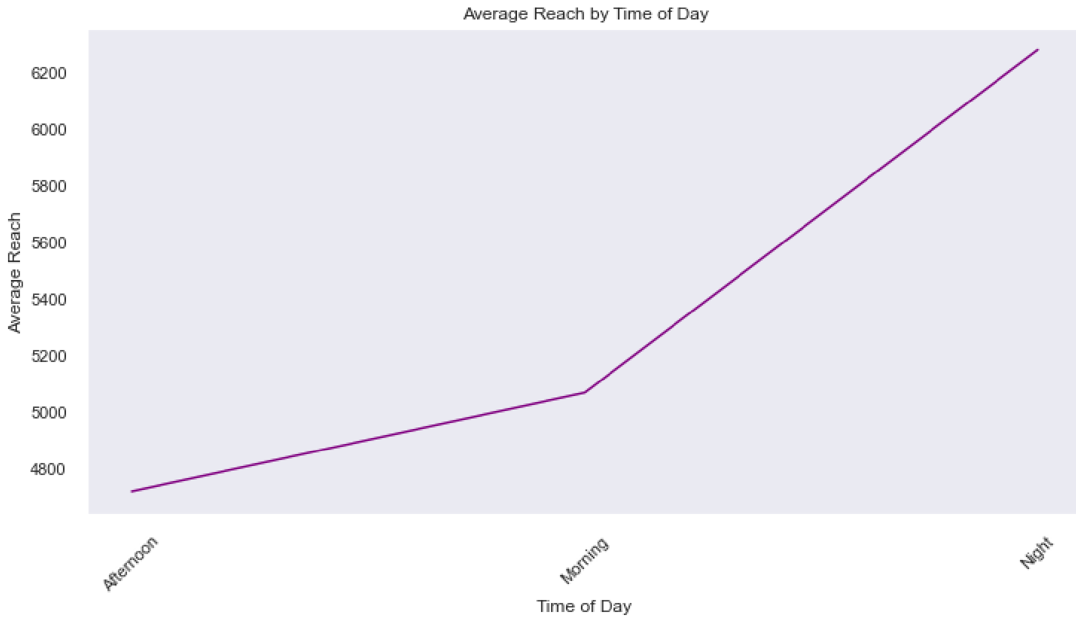
Afternoon Dip: A noticeable dip in engagement is observed during the afternoon hours. This midday decline in engagement levels could be attributed to various factors, such as users being occupied with work or other activities during this time. Understanding this daily engagement pattern can inform your content strategy. You might consider strategically scheduling your most engaging or important posts for the morning and evening periods when user activity and interest are at their highest. Conversely, during the afternoon lull, you could focus on less critical content or use this time for analysis and planning. By aligning your posting schedule with these engagement trends, you can optimize the impact of your social media efforts.

In [64]:

```python
# Plot reach by time of day
plt.figure(figsize=(12, 6))
reach_by_time.plot(kind='line', color='purple')
plt.title('Average Reach by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Average Reach')
plt.xticks(rotation=45)
plt.grid()

plt.show()
```

Average Reach by Time of Day

In [64]:
```
# Plot reach by time of day
plt.figure(figsize=(12, 6))
reach_by_time.plot(kind='line', color='purple')
plt.title('Average Reach by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Average Reach')
plt.xticks(rotation=45)
plt.grid()

plt.show()
```



The observed pattern in reach closely mirrors the engagement trends, indicating a strong correlation between the two. Here's a more refined context for this observation:

Parallel Reach and Engagement: It's evident that the reach of your posts follows a similar pattern to that of engagement, displaying a robust correlation with the hour of the day. Just like with engagement, reach experiences its peaks during the latter hours of the day.

Concurrent Peaks: The fact that both reach and engagement reach their respective zeniths during the same timeframe further emphasizes the synchronicity of user interactions with your content. This alignment suggests that your posts effectively capture and maintain the attention of your audience during these peak hours.

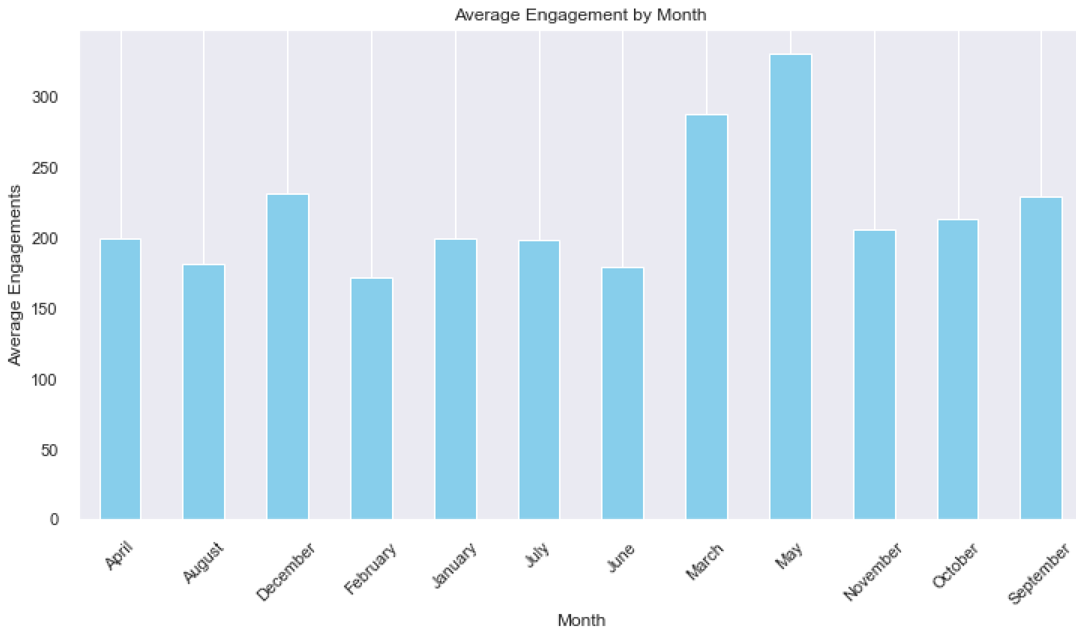
This synergy between reach and engagement hours highlights the importance of strategic timing. Capitalizing on these peak periods can significantly enhance the visibility and impact of your posts. By recognizing and optimizing around these trends, you can potentially amplify your social media reach and engagement to even greater heights.

In [65]:
```
# Explore engagement by month
engagement_by_month = df1.groupby('Month')['Engagements'].mean()
```

In [66]:
```
# Explore user activity (e.g., comments) by month
activity_by_month = df1.groupby('Month')['Comments'].mean()
```

In [67]:
```
# Plot engagement by month
plt.figure(figsize=(12, 6))
engagement_by_month.plot(kind='bar', color='skyblue')
plt.title('Average Engagement by Month')
plt.xlabel('Month')
plt.ylabel('Average Engagements')
plt.xticks(rotation=45)
plt.grid(axis='y')
```

In [67]:
```python
# Plot engagement by month
plt.figure(figsize=(12, 6))
engagement_by_month.plot(kind='bar', color='skyblue')
plt.title('Average Engagement by Month')
plt.xlabel('Month')
plt.ylabel('Average Engagements')
plt.xticks(rotation=45)
plt.grid(axis='y')
```



The chart vividly illustrates the dynamic changes in engagement levels across different months, with distinct variations that are worth noting. Here's an enhanced context for this observation:

Monthly Engagement Dynamics: The data reveals a compelling variance in engagement levels throughout the year, with some months standing out prominently. Notably, the months of May and March emerge as clear frontrunners, characterized by a substantial surge in user interactions and engagements with your content.

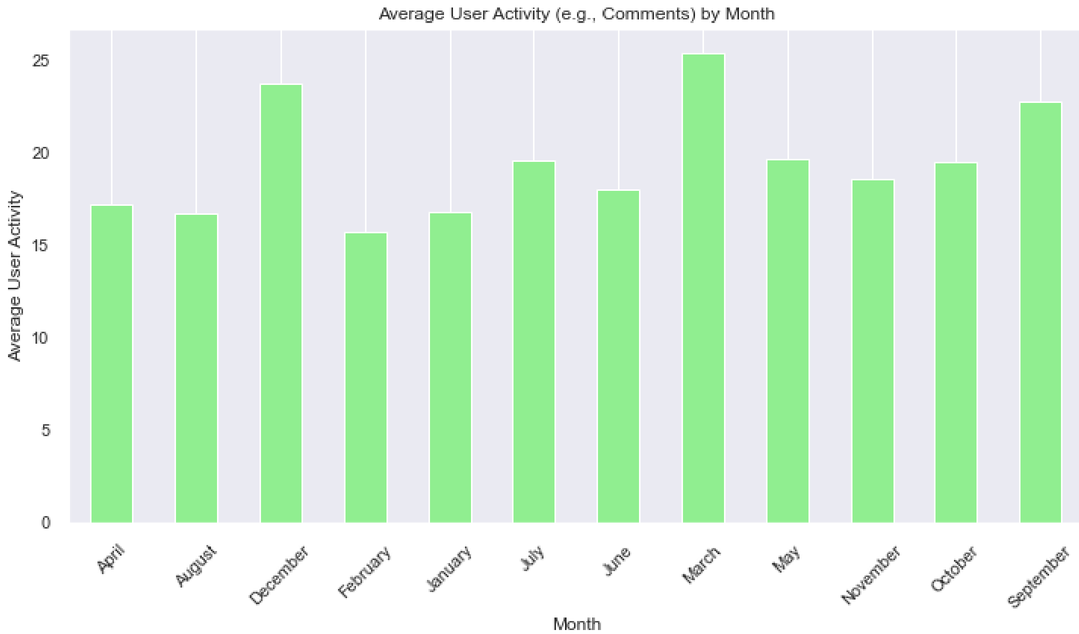
Low Engagement Months: Conversely, February, August, and June exhibit a noticeable dip in engagement frequency. These months consistently record the lowest engagement metrics, indicating that user interactions with your posts tend to be comparatively subdued during these periods.

These pronounced fluctuations in engagement across the months underline the importance of seasonality in your social media strategy. Understanding these trends can guide your content planning and posting schedule. For instance, during high-engagement months like May and March, you might consider allocating more resources and focus to capitalize on the heightened user interest. Conversely, in lower-engagement months, you can strategize ways to maintain user engagement or explore alternative approaches to drive interactions with your content.

In [68]:
```python
# Plot user activity (e.g., comments) by month
plt.figure(figsize=(12, 6))
activity_by_month.plot(kind='bar', color='lightgreen')
plt.title('Average User Activity (e.g., Comments) by Month')
plt.xlabel('Month')
plt.ylabel('Average User Activity')
plt.xticks(rotation=45)
plt.grid(axis='y')

plt.show()
```

Average User Activity (e.g., Comments) by Month

In [68]:
```
# Plot user activity (e.g., comments) by month
plt.figure(figsize=(12, 6))
activity_by_month.plot(kind='bar', color='lightgreen')
plt.title('Average User Activity (e.g., Comments) by Month')
plt.xlabel('Month')
plt.ylabel('Average User Activity')
plt.xticks(rotation=45)
plt.grid(axis='y')

plt.show()
```



The analysis of comments across the months unveils a striking pattern of variation, with certain months clearly standing out in terms of user activity:

Monthly Comment Dynamics: The data highlights significant fluctuations in user comments throughout the year, with specific months demonstrating distinct levels of activity. Notably, the months of December, March, and September emerge as the frontrunners, characterized by a substantial surge in user comments and interaction with your content.

Lower Comment Months: In contrast, January, February, August, and April exhibit markedly lower comment frequency. These months consistently record the fewest user comments, indicating that user engagement with the content tends to be more subdued during these periods.

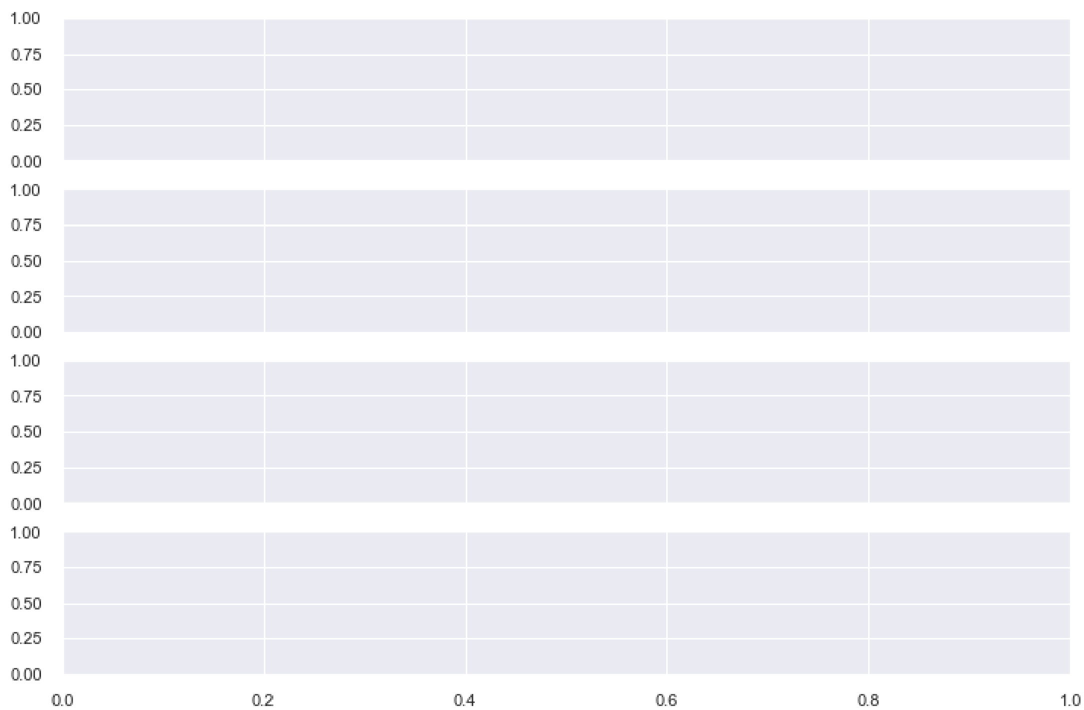
Understanding these comment trends across the months is instrumental in shaping your social media strategy. During high-comment months like December, March, and September, you may consider fostering and promoting discussions, as your audience appears to be more actively engaged. Conversely, in lower-comment months, exploring strategies to encourage and stimulate user interactions with your posts becomes particularly important.

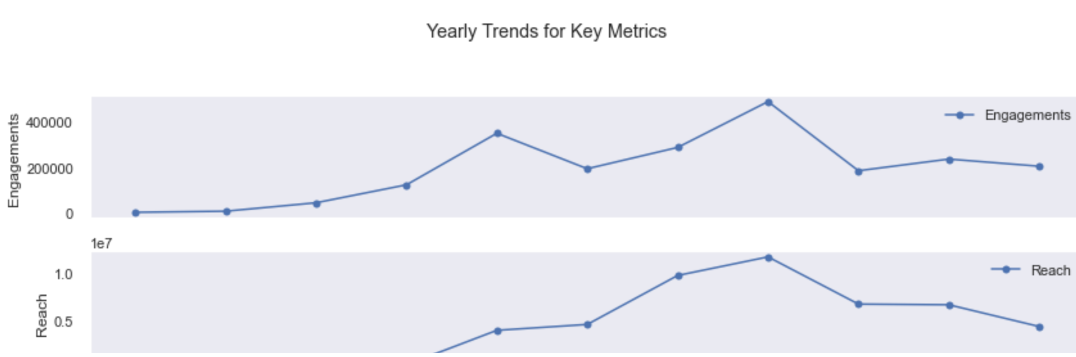
In [69]:
```
# Define the key metrics to analyze over the years
metrics_to_analyze = ['Engagements', 'Reach', 'Comments', 'Shares']
```

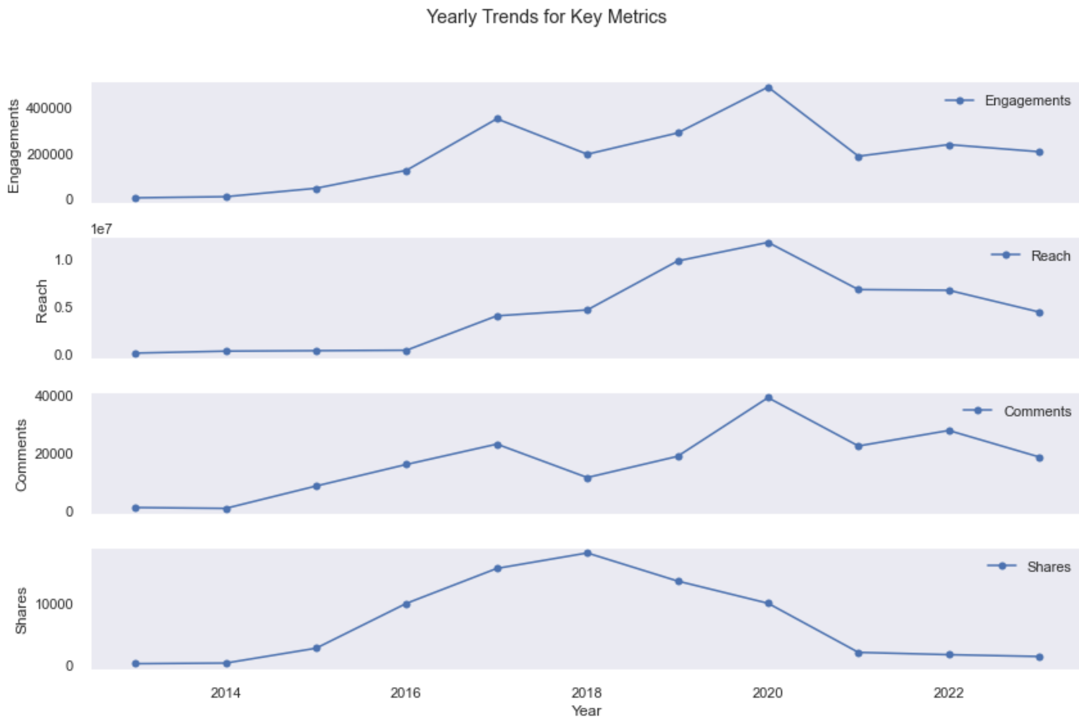
In [70]:
```
# Create subplots for each metric
fig, axes = plt.subplots(len(metrics_to_analyze), 1, figsize=(12, 8), sharex=True)

# Create subplots for each metric
fig, axes = plt.subplots(len(metrics_to_analyze), 1, figsize=(12, 8), sharex=True)

# Plot each metric over the years
for i, metric in enumerate(metrics_to_analyze):
    ax = axes[i]
    metric_data = df1.groupby('Year')[metric].sum()
    ax.plot(metric_data.index, metric_data.values, marker='o', linestyle='-', label=met
    ax.set_ylabel(metric)
```

In [70]:

```python
# Create subplots for each metric
fig, axes = plt.subplots(len(metrics_to_analyze), 1, figsize=(12, 8), sharex=True)

# Create subplots for each metric
fig, axes = plt.subplots(len(metrics_to_analyze), 1, figsize=(12, 8), sharex=True)

# Plot each metric over the years
for i, metric in enumerate(metrics_to_analyze):
    ax = axes[i]
    metric_data = df1.groupby('Year')[metric].sum()
    ax.plot(metric_data.index, metric_data.values, marker='o', linestyle='-', label=met
    ax.set_ylabel(metric)
    ax.grid()
    ax.legend()

# Set the common x-axis label and title
plt.xlabel('Year')
plt.suptitle('Yearly Trends for Key Metrics')

# Display the plots
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```







The analysis of key metrics over the years reveals compelling insights into the evolving patterns of user engagement and behavior:

Engagements: The engagement metric exhibited an intriguing trajectory. It initially showed an upward trend in 2014, experiencing consistent growth until it peaked in 2017. Subsequently, there was a decline in engagements, followed by another significant peak in 2020. This surge in 2020 raises the question of whether the global events, such as the COVID-19 pandemic, which led to people spending more time at home, played a role in driving higher engagement rates.

Reach: The reach metric displayed a distinctive pattern. It began to show noticeable growth in 2016, eventually reaching its zenith in 2020. After this peak, there was a subsequent decline in reach. This trend in reach metrics suggests a changing landscape in terms of how content is disseminated and received.

Comments: Comments on posts followed an intriguing pattern as well. They experienced an initial upward trend in 2014, reaching their first peak in 2017. Following this peak, there was a period of declining comments. However, it's noteworthy that comments surged to another peak in 2020. This dual-peak pattern in comments prompts further exploration to understand the factors influencing user interactions.

Shares: The metric of shares also underwent a notable transformation. It demonstrated an upward trend in 2014, achieving its highest point in 2018. Subsequently, there was a decline in the frequency of shares. The shifting dynamics of shares hint at changing user behaviors and preferences in sharing content.

These observations underscore the dynamic nature of user engagement and interaction over the years. Factors such as global events and changes in online behavior may have contributed to these fluctuations. Investigating the underlying drivers behind these trends can provide valuable insights for adapting your social media strategy and effectively engaging with your audience.

In [72]:
```
# Define the columns to analyze
columns_to_analyze = ['Content Type', 'Engagements', 'Impressions', 'Reach', 'Comments'
```

In [74]:
```
# Create subplots
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 8))
```

In [73]:
```
# Group the data by 'Content Type'
content_type_grouped = df1.groupby('Content Type')[columns_to_analyze].mean()
```

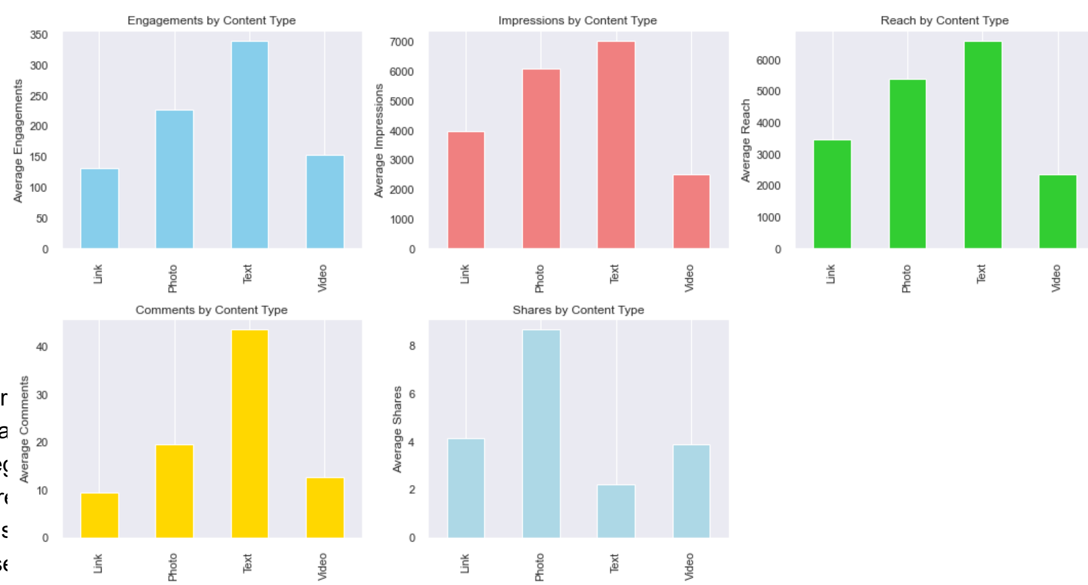
```
# Plot Engagements
content_type_grouped['Engagements'].plot(kind='bar', ax=axes[0, 0], color='skyblue')
axes[0, 0].set_xlabel('')
axes[0, 0].set_ylabel('Average Engagements')
axes[0, 0].set_title('Engagements by Content Type')
axes[0, 0].grid(axis='y')

# Plot Impressions
content_type_grouped['Impressions'].plot(kind='bar', ax=axes[0, 1], color='lightcoral')
```

In [74]:

```
# Create subplots
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 8))
```

In [73]:

```
# Group the data by 'Content Type'
content_type_grouped = df1.groupby('Content Type')[columns_to_analyze].mean()
# Plot Engagements
content_type_grouped['Engagements'].plot(kind='bar', ax=axes[0, 0], color='skyblue')
axes[0, 0].set_xlabel('')
axes[0, 0].set_ylabel('Average Engagements')
axes[0, 0].set_title('Engagements by Content Type')
axes[0, 0].grid(axis='y')

# Plot Impressions
content_type_grouped['Impressions'].plot(kind='bar', ax=axes[0, 1], color='lightcoral')
axes[0, 1].set_xlabel('')
axes[0, 1].set_ylabel('Average Impressions')
axes[0, 1].set_title('Impressions by Content Type')
axes[0, 1].grid(axis='y')

# Plot Reach
content_type_grouped['Reach'].plot(kind='bar', ax=axes[0, 2], color='limegreen')
axes[0, 2].set_xlabel('')
axes[0, 2].set_ylabel('Average Reach')
axes[0, 2].set_title('Reach by Content Type')
axes[0, 2].grid(axis='y')

# Plot Comments
content_type_grouped['Comments'].plot(kind='bar', ax=axes[1, 0], color='gold')
axes[1, 0].set_xlabel('')
axes[1, 0].set_ylabel('Average Comments')
axes[1, 0].set_title('Comments by Content Type')
axes[1, 0].grid(axis='y')

# Plot Shares
content_type_grouped['Shares'].plot(kind='bar', ax=axes[1, 1], color='lightblue')
axes[1, 1].set_xlabel('')
axes[1, 1].set_ylabel('Average Shares')
axes[1, 1].set_title('Shares by Content Type')
axes[1, 1].grid(axis='y')

# Remove the empty subplot
fig.delaxes(axes[1, 2])

# Adjust spacing between subplots
plt.tight_layout()

# Show the plots
plt.show()
```

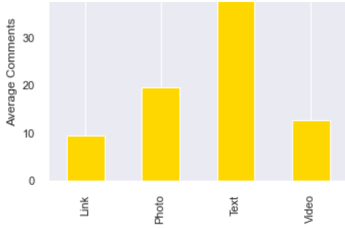


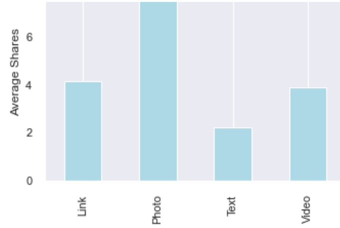
After ... enga... categ... share... this s... these... the dominance of text and photo content across all analyzed metrics, emphasizing their effectiveness in engaging the audience.

In [26]:

```
df1.head(1)
```

After
enga
categ
share
this s
these





the dominance of text and photo content across all analyzed metrics, emphasizing their effectiveness in engaging the audience.

In [26]:
```python
df1.head(1)
```

Out[26]:

Non-viral Reach	Engaged Fans	Users Talking About This	Unique Post Clicks	Unique Reactions	Unique Comments	Unique Shares	Fan Reach	Engaged Users	Viral Impressions	Non-viral Impressions
627.0	51.0	231.0	587.0	177.0	53.0	2.0	1112.0	792.0	5.0	207373.0

In [71]:
```python
#create a list of variables to test the relationship against Engagements
df2 = df1[['Engagements','Comments','Shares','Click-Through Rate','Negative Feedback']]
```

In [72]:
```python
df2 = df2.reset_index(drop=True)
```

In [73]:
```python
# Define the dependent variable (Y) and independent variables (X)
Y = df2['Engagements']
X = df2.drop(columns=['Engagements'])
```

In [74]:
```python
# Add a constant term to the independent variables
X = sm.add_constant(X)

# Fit the regression model
model = sm.OLS(Y, X).fit()

# Print the model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Engagements   R-squared:                       0.524
```

In [74]:

```python
# Add a constant term to the independent variables
X = sm.add_constant(X)

# Fit the regression model
model = sm.OLS(Y, X).fit()

# Print the model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Engagements   R-squared:                       0.524
Model:                            OLS   Adj. R-squared:                  0.524
Method:                 Least Squares   F-statistic:                     2626.
Date:                Tue, 03 Oct 2023   Prob (F-statistic):               0.00
Time:                        12:02:04   Log-Likelihood:                -76629.
No. Observations:                9553   AIC:                         1.533e+05
Df Residuals:                    9548   BIC:                         1.533e+05
Df Model:                           4
Covariance Type:            nonrobust
=====================================================================================
=
                        coef    std err          t      P>|t|      [0.025      0.97
5]
-------------------------------------------------------------------------------------
-
const              -109.9693      8.631    -12.741      0.000    -126.888     -93.05
0
Comments              8.7479      0.153     57.037      0.000       8.447       9.04
9
Shares               22.9670      0.458     50.158      0.000      22.069      23.86
5
Click-Through Rate   15.1782    115.876      0.131      0.896    -211.963     242.32
0
Negative Feedback   -67.7558     10.583     -6.402      0.000     -88.500     -47.01
1
==============================================================================
Omnibus:                    26832.086   Durbin-Watson:                   1.927
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       1993495296.968
Skew:                          36.281   Prob(JB):                         0.00
Kurtosis:                    2239.737   Cond. No.                         934.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
```

R-squared Value: The R-squared value of 0.524 indicates that the model explains approximately 52.4% of the variance in user engagements. In other words, the model reasonably captures the variation in engagement levels based on the selected independent variables.

Coefficients:

Comments (Positive Coefficient): The coefficient of 8.7479 for Comments suggests that an increase in the number of comments on a Facebook post is associated with a significant positive impact on user engagements. Specifically, for every additional comment, we can expect user engagements to increase by approximately 8.75 units.

Shares (Positive Coefficient): The coefficient of 22.9670 for Shares indicates that an increase in the number of shares a post receives also has a substantial positive influence on user engagements. For each additional share, user engagements are predicted to increase by about 22.97 units.

Negative Feedback (Negative Coefficient): The coefficient of -67.7558 for Negative Feedback reveals a significant negative relationship. This means that an increase in negative feedback on a post is associated with a decrease in user engagements. Each unit increase in negative feedback is linked to a decrease of approximately 67.76 units in user engagements.

Click-Through Rate (CTR) (Insignificant Coefficient): The coefficient of 15.1782 for CTR appears to be statistically insignificant (p-value of 0.896), suggesting that changes in the click-through rate do not have a substantial impact on user engagements in this model.

Time Series Analysis for Engagements

In [214]:

```python
df3 = df[['Date', 'Engagements']]
df3.columns = ['ds', 'y']  # Rename columns to 'ds' and 'y' for Prophet
```

Shares (Positive Coefficient): The coefficient of 22.9670 for Shares indicates that an increase in the number of shares a post receives also has a substantial positive influence on user engagements. For each additional share, user engagements are predicted to increase by about 22.97 units.

Negative Feedback (Negative Coefficient): The coefficient of -67.7558 for Negative Feedback reveals a significant negative relationship. This means that an increase in negative feedback on a post is associated with a decrease in user engagements. Each unit increase in negative feedback is linked to a decrease of approximately 67.76 units in user engagements.

Click-Through Rate (CTR) (Insignificant Coefficient): The coefficient of 15.1782 for CTR appears to be statistically insignificant (p-value of 0.896), suggesting that changes in the click-through rate do not have a substantial impact on user engagements in this model.

Time Series Analysis for Engagements

In [214]:
```
df3 = df[['Date', 'Engagements']]
df3.columns = ['ds', 'y']  # Rename columns to 'ds' and 'y' for Prophet
```

In [215]:
```
df3.head()
```

Out[215]:

	ds	y
0	12/17/2022 5:08 pm	1024.0
1	2019-04-05 10:01:00	5876.0
2	2020-02-06 21:00:00	4744.0
3	2022-09-05 10:37:00	445.0
4	7/18/2021 10:00 am	4730.0

In [216]:
```
#convert the ds column to date format
df3['ds'] = pd.to_datetime(df3['ds']).dt.date
```

In [217]:
```
# Initialize the Prophet model
model = Prophet()
```

In [218]:
```
# Fit the model to your data
model.fit(df3)
```

```
15:54:51 - cmdstanpy - INFO - Chain [1] start processing
15:54:53 - cmdstanpy - INFO - Chain [1] done processing
```

Out[218]: <prophet.forecaster.Prophet at 0x20a730dfeb0>

In [219]:
```
# Create a future dataframe for forecasting
future = model.make_future_dataframe(periods=365)
```

In [220]:
```
# Make predictions
forecast = model.predict(future)
```

In [221]:
```
forecast.tail(5)
```

Out[221]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_term
3063	2024-12-02	376.697362	-1120.269381	1738.212044	249.341023	492.417292	-59.485611	-59
3064	2024-12-03	376.952373	-1014.886665	1843.207836	249.399880	493.499121	-55.549163	-55
3065	2024-12-04	377.207384	-1211.110942	1769.859221	249.914330	494.556465	-67.575059	-67
3066	2024-12-05	377.462394	-1137.519827	1676.950617	249.760173	495.296869	9.071640	9
3067	2024-12-06	377.717405	-1087.155591	1715.723730	249.256296	496.037272	-47.916651	-47

In [224]:
```
forecast[['ds', 'yhat']]
```

Out[224]:

	ds	yhat
0	2013-01-02	136.792810
1	2013-01-15	107.185904
2	2013-01-16	98.943975
3	2013-01-17	179.647984
4	2013-02-04	179.267519

3066	2024-12-05	377.462394	-1137.519827	1676.950617	249.760173	495.296869	9.071640	9
3067	2024-12-06	377.717405	-1087.155591	1715.723730	249.256296	496.037272	-47.916651	-47

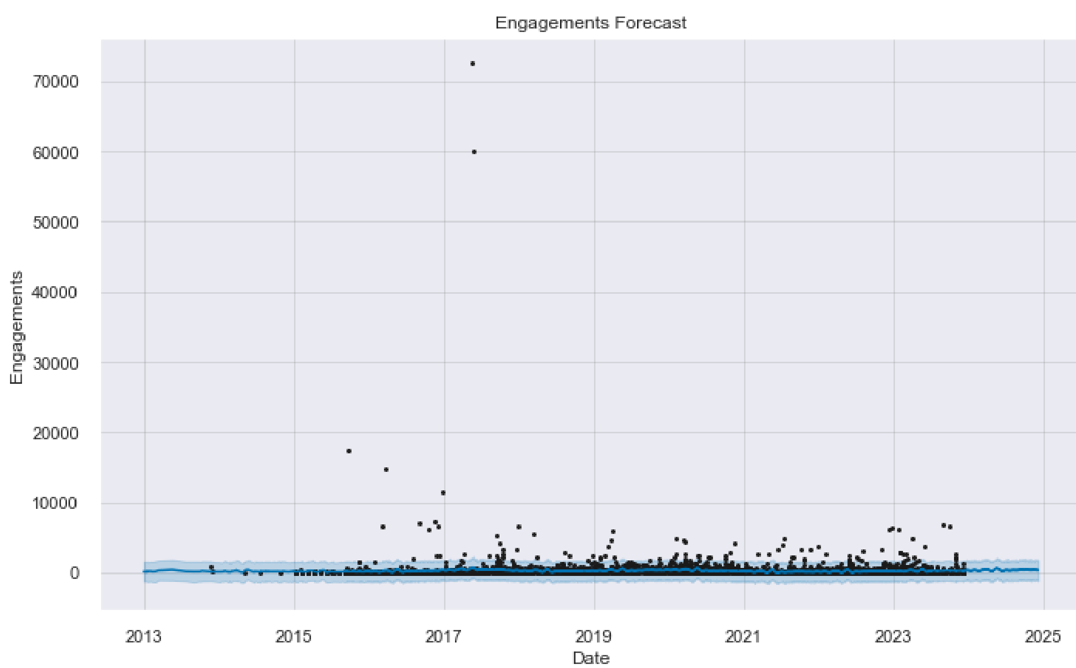
In [224]:
```
forecast[['ds', 'yhat']]
```

Out[224]:

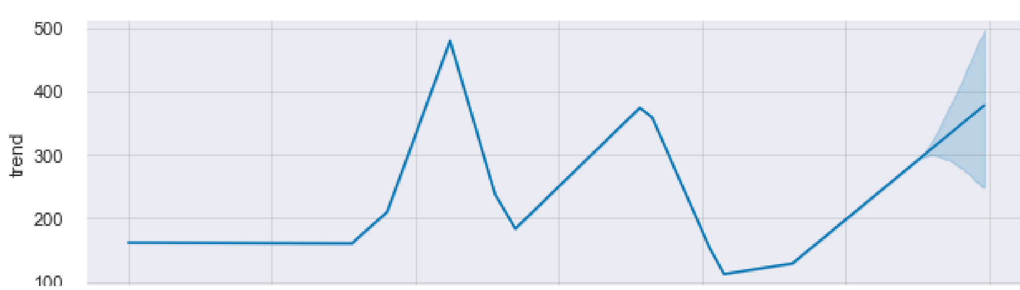
	ds	yhat
0	2013-01-02	136.792810
1	2013-01-15	107.185904
2	2013-01-16	98.943975
3	2013-01-17	179.647984
4	2013-02-04	179.267519
...	...	...
3063	2024-12-02	317.211752
3064	2024-12-03	321.403210
3065	2024-12-04	309.632325
3066	2024-12-05	386.534034
3067	2024-12-06	329.800753

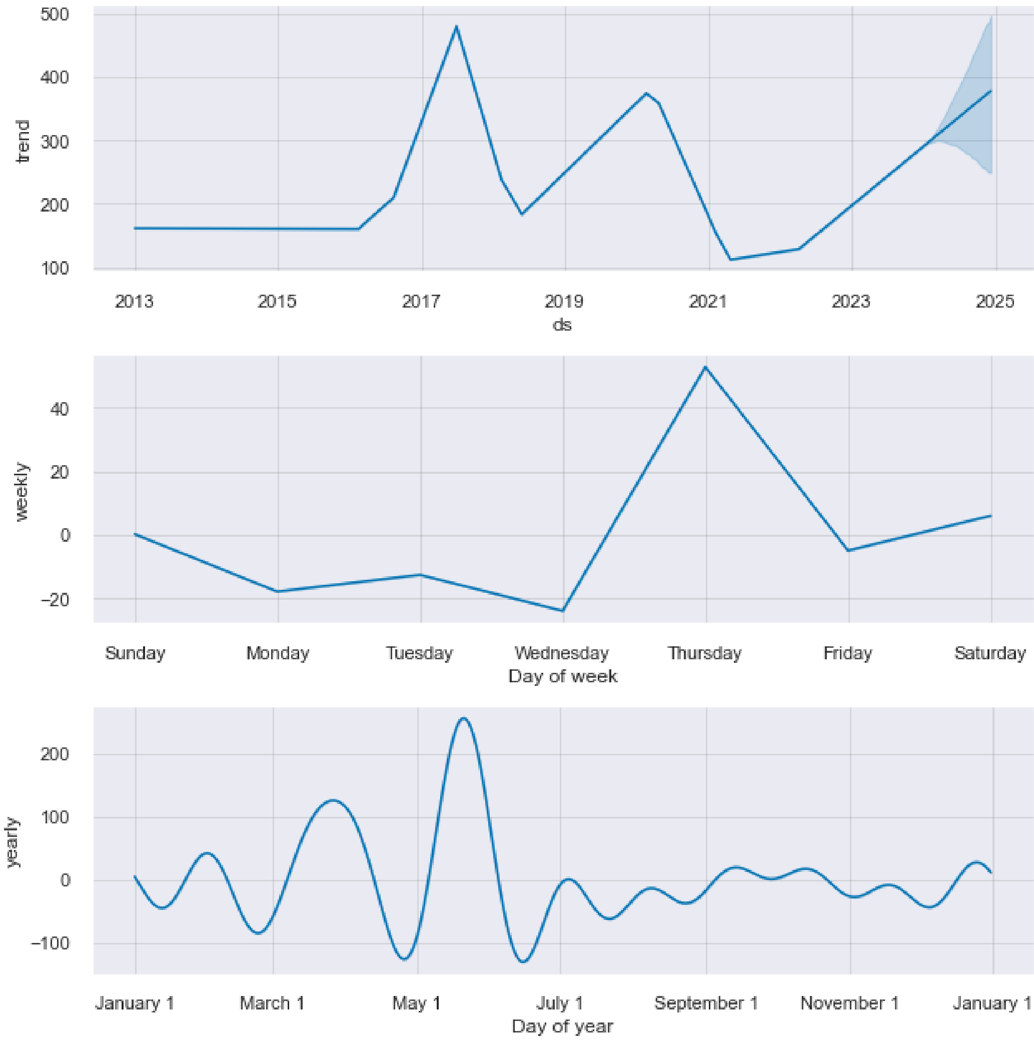
3068 rows × 2 columns

In [222]:
```
# Plot the forecast
fig = model.plot(forecast)
plt.title('Engagements Forecast')
plt.xlabel('Date')
plt.ylabel('Engagements')
plt.show()
```



In [223]:
```
fig2 = model.plot_components(forecast)
```



In [223]:
```
fig2 = model.plot_components(forecast)
```


In [ ]:
```

```

SUMMARY OF FINDINGS

The average negative feedback received per post is approximately 0.298, with a wide range from 0 to a maximum of 13. Negative feedback may include reactions like "angry" or "dislike."

On average, posts were displayed to users around 5,712 times. However, there is a significant variation, ranging from 0 to over 207,000 impressions.

The average engagement per post is about 220.91

On average, posts received approximately 19.47 comments. There is considerable variation, ranging from no comments to over 1,500

Posts were shared an average of 7.87 times. However, the number of shares varies widely, with some posts having no shares and others up to 588 shares.

The analysis of content types on Facebook posts uncovers a clear trend: posts containing photos consistently garner the highest levels of user engagement, including diverse types of reactions. This pattern corresponds with the dataset's makeup, where a substantial portion of posts consists of photos. These findings highlight the effectiveness of incorporating visual elements, particularly photos, into your content strategy for boosting user engagement on Facebook. This strategy not only capitalizes on the dataset's existing content composition but also aligns with the broader industry trend emphasizing the power of visual content in capturing and resonating with online audiences.

The average CTR is very low at 0.0018, indicating that a small percentage of users clicked on links within the posts.

The analysis of user interactions with your Facebook posts reveals that certain emotional reactions, specifically "Love," "Sad," "Angry," and "Haha" reactions, have generated some of the highest frequencies among the various types of user engagement. These reactions offer valuable insights into how your audience emotionally connects with your content.

User comments on Facebook posts show a positive correlation with both shares and overall engagement, indicating that posts with more comments tend to garner increased sharing and engagement.

Posts were shared an average of 7.87 times. However, the number of shares varies widely, with some posts having no shares and others up to 588 shares.

The average CTR is very low at 0.0016, indicating that a small percentage of users clicked on links within the posts.

The analysis of content types on Facebook posts uncovers a clear trend: posts containing photos consistently garner the highest levels of user engagement, including diverse types of reactions. This pattern corresponds with the dataset's makeup, where a substantial portion of posts consists of photos. These findings highlight the effectiveness of incorporating visual elements, particularly photos, into your content strategy for boosting user engagement on Facebook. This strategy not only capitalizes on the dataset's existing content composition but also aligns with the broader industry trend emphasizing the power of visual content in capturing and resonating with online audiences.

The analysis of user interactions with your Facebook posts reveals that certain emotional reactions, specifically "Love," "Sad," "Angry," and "Haha" reactions, have generated some of the highest frequencies among the various types of user engagement. These reactions offer valuable insights into how your audience emotionally connects with your content.

User comments on Facebook posts show a positive correlation with both shares and overall engagement, indicating that posts with more comments tend to garner increased sharing and engagement.

User engagement on Facebook posts follows a weekly pattern, with higher levels on Saturdays, Sundays, and Fridays, while Thursdays also show increased activity. In contrast, engagement is lower on Mondays, Tuesdays, and Wednesdays, indicating a connection between user activity and the days of the week. This information can guide your content scheduling strategy, focusing on peak engagement days for more impactful posts.

The analysis of reach across the week indicates consistent reach levels every day, highlighting ongoing audience engagement. Notably, Saturdays and Sundays exhibit significant spikes in reach, emphasizing their importance in your content strategy for maximizing post impact and audience reach over the weekend.

The analysis of daily engagement patterns reveals a distinct trend: engagement levels steadily rise from morning to nighttime, peaking in the evening. However, there is a noticeable dip in engagement during the afternoon, likely due to users' focus on other activities.

The analysis of reach patterns throughout the day reveals a strong correlation with engagement trends, indicating parallel peaks during the later hours. This synchronization underscores the effectiveness of capturing audience attention during these peak periods, emphasizing the importance of strategic timing in content posting.

The analysis of monthly engagement patterns underscores significant variations throughout the year. May and March consistently stand out as high-engagement months, while February, August, and June record lower engagement levels. These findings highlight the influence of seasonality on user interactions and suggest the need for adaptive content strategies that align with these trends.

The analysis of monthly comment patterns reveals significant fluctuations in user activity. December, March, and September stand out as high-comment months, while January, February, August, and April consistently record lower comment frequencies. Engagements: Engagements initially rose in 2014, peaking in 2017 before dropping. A significant surge occurred in 2020, possibly influenced by the COVID-19 pandemic and increased time spent at home.

Reach: Reach started growing in 2016, peaked in 2020, and declined afterward, reflecting shifts in content dissemination. Comments: Comments followed a dual-peak pattern, with peaks in 2017 and 2020, suggesting evolving user interactions. Shares: Shares increased until 2018, followed by a decline, reflecting changing sharing behaviors.

The analysis of content types, including text, photo, and video, consistently highlights text and photo content as top performers across various engagement metrics. Whether considering average engagements, impressions, reach, comments, or shares, these content types consistently demonstrate strong performance.

The regression analysis highlights the significant positive influence of Comments and Shares on user engagements on Facebook. Posts with more comments and shares tend to generate higher levels of engagement. Conversely, Negative Feedback has a substantial negative impact on user engagements. The Click-Through Rate, in this particular analysis, does not appear to be a significant predictor of user

RECOMMENDATION

Based on the summary of findings from the Facebook analysis, here are detailed recommendations to optimize your Facebook content strategy:

Leverage Emotional Reactions: Capitalize on the emotional connections your audience has with your content. Continue creating posts that elicit "Love," "Sad," "Angry," and "Haha" reactions, as these emotions drive high engagement. Craft content that resonates with these emotional triggers, encouraging users to express themselves through reactions.

Visual Content Dominance: Recognize the power of visual content, particularly photos, in driving user engagement. Given that photo posts consistently perform the best, consider incorporating more visual elements into your content strategy. Utilize high-quality images, infographics, and visually appealing graphics to capture and maintain audience attention.

Encourage Comments: Foster discussions and conversations on your Facebook posts, as user comments positively correlate with overall engagement. Encourage user-generated content, ask questions, and prompt feedback to increase comment-driven interactions. Respond promptly to comments to further stimulate engagement.

RECOMMENDATION

Based on the summary of findings from the Facebook analysis, here are detailed recommendations to optimize your Facebook content strategy:

Leverage Emotional Reactions: Capitalize on the emotional connections your audience has with your content. Continue creating posts that elicit "Love," "Sad," "Angry," and "Haha" reactions, as these emotions drive high engagement. Craft content that resonates with these emotional triggers, encouraging users to express themselves through reactions.

Visual Content Dominance: Recognize the power of visual content, particularly photos, in driving user engagement. Given that photo posts consistently perform the best, consider incorporating more visual elements into your content strategy. Utilize high-quality images, infographics, and visually appealing graphics to capture and maintain audience attention.

Encourage Comments: Foster discussions and conversations on your Facebook posts, as user comments positively correlate with overall engagement. Encourage user-generated content, ask questions, and prompt feedback to increase comment-driven interactions. Respond promptly to comments to further stimulate engagement.

Strategic Content Scheduling: Optimize your content posting schedule based on weekly engagement patterns. Focus on Saturdays, Sundays, and Fridays when engagement is typically higher. Plan important posts, campaigns, or promotions for these peak engagement days. Use Mondays, Tuesdays, and Wednesdays for lighter content or behind-the-scenes updates.

Weekend Reach Strategy: Maximize post impact and reach by concentrating on weekends, particularly Saturdays and Sundays, when reach experiences significant spikes. Strategically time your posts to align with these high-reach periods to expand your audience and content visibility.

Time-Sensitive Posting: Recognize the daily engagement trends, with engagement steadily rising from morning to nighttime. Schedule your most critical or engaging posts for the evening when users are most active and attentive. Be mindful of the afternoon dip and adjust your content strategy accordingly.

Strategic Timing for Reach: Align the timing of your content posts with engagement trends. Focus on capturing audience attention during the evening and nighttime hours when both engagement and reach peak. Timing your posts strategically can enhance content visibility and user interactions.

Seasonal Adaptation: Acknowledge the impact of seasonality on user interactions and engagement. Develop adaptive content strategies that align with seasonal trends, emphasizing high-engagement months like May and March. Tailor your content themes, topics, and promotions to resonate with the audience during specific seasons.

Content Type Optimization: Continue prioritizing text and photo content, as they consistently perform well across various engagement metrics. Balance your content mix by featuring these content types prominently, as they resonate effectively with your audience. When using videos or other formats, ensure they complement your core text and photo content.

Focus on Reactions and Comments: Recognize the importance of "Reactions" and "Comments" in driving overall engagement. Continue to encourage user reactions and discussions, as these metrics have the most substantial positive impact on overall interactions. While "Impressions," "Shares," and "Reach" contribute positively, prioritize "Reactions" and "Comments" to maximize engagement.

Continuous Monitoring: Maintain a proactive approach to your content strategy by continually monitoring engagement metrics and adjusting your tactics based on trends. Experiment with content formats, messaging, and posting times to optimize your strategy for evolving user behavior.

Multichannel Consideration: Explore the possibility of diversifying your online presence beyond Facebook, considering other social media platforms where your audience may be active. Expanding your digital footprint can help you reach a broader audience and increase overall engagement.

By implementing these recommendations, you can refine your Facebook content strategy, enhance user engagement, and foster a vibrant and active online community.

```
In [ ]:
```