Data Analysis Assessment

Christina(Yiting) Zhang

1. What is the typical engagement rate we can expect? What's the likelihood that we can achieve a 15% engagement rate?
2. Does day of the week and time of posting affect engagement rates?
3. How are our game titles doing in terms of social performance? Is there a specific game we should focus more on or less?
4. What media type performs the best?
5. What is our best performing campaign?
6. Define out a posting strategy for our social channels based on your discoveries.
7. What suggestions would you give to the social media team if they want to expand their presence (e.g. if our CSGO youtube channel is doing well should we expand to TikTok)?

Importing Data

In [7]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from plotly import express as px
```

In [8]:
```
df = pd.read_csv('social_data.csv')
df.head()
```

Out[8]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type
0	03-31-2023 19:55	CSGO	TWITTER	N/A	0	0	Text
1	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text
2	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text
3	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text
4	03-31-2023 19:43	CSGO	TWITTER	Community Engagement	9517	1215	Video

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3479 entries, 0 to 3478
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Published Date     3479 non-null   object
 1   Account            3479 non-null   object
 2   Account Type       3479 non-null   object
 3   Campaign Name      3479 non-null   object
 4   Total Impressions  3479 non-null   int64
 5   Total Engagements  3479 non-null   int64
 6   Media Type         3479 non-null   object
dtypes: int64(2), object(5)
memory usage: 190.4+ KB
```

In [4]:
```
df.describe()
```

Out[4]:

	Total Impressions	Total Engagements
count	3479.000000	3479.000000
mean	10972.453579	947.390629
std	29577.108063	3468.628038
min	0.000000	0.000000
25%	167.000000	0.000000
50%	1884.000000	39.000000
75%	12467.000000	276.500000
max	753227.000000	95062.000000

In [3]:
```
df['Account Type'].unique()
```

Out[3]:
```
array(['TWITTER', 'FBPAGE', 'INSTAGRAM', 'TIKTOK_BUSINESS', 'YOUTUBE',
       'LINKEDIN_COMPANY'], dtype=object)
```

1. What is the typical engagement rate we can expect? What's the likelihood that we can achieve a 15% engagement rate?

First, we should look at the engagements and see if there is any pattern.

In [28]:
```
fig = px.box(df, y = "Total Engagements",
             height=500,width=600,
             title='Distribution of Total Engagement')

fig.show()
```

In [11]:
```
df['Total Engagements'].describe()
```

Out[11]:
```
count     3479.000000
mean       947.390629
std       3468.628038
min          0.000000
25%          0.000000
50%         39.000000
75%        276.500000
max      95062.000000
Name: Total Engagements, dtype: float64
```

A typical engagement we can expect is around 947 across all social media platforms.

In [10]:
```
df.groupby('Account Type')['Total Engagements'].describe()
```

Out[10]:

Account Type	count	mean	std	min	25%	50%	75%	max
FBPAGE	585.0	4405.613675	7317.270756	0.0	0.00	1687.0	5994.0	95062.0
INSTAGRAM	588.0	143.005102	538.534299	0.0	0.00	0.0	6.0	5472.0
LINKEDIN_COMPANY	22.0	38.590909	37.700144	0.0	16.25	28.5	41.5	133.0
TIKTOK_BUSINESS	113.0	166.946903	939.566065	0.0	11.00	40.0	83.0	9680.0
TWITTER	1951.0	309.871348	974.696962	0.0	9.00	51.0	223.5	12348.0
YOUTUBE	220.0	46.945455	299.680190	0.0	0.00	9.5	35.0	4388.0

We expect most engagement from Facebook and little engagement from LinkedIn and Youtube.

Now let's calculate the `Engagement Rate`.

- Engagement rate=Engagement / Impression*100

In [25]:
```
df['Engagement Rate']=df['Total Engagements']/df['Total Impressions']*100

df.head()
```

Out[25]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate
0	03-31-2023 19:55	CSGO	TWITTER	N/A	0	0	Text	NaN
1	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	NaN
2	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	NaN
3	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	NaN
4	03-31-2023 19:43	CSGO	TWITTER	Community Engagement	9517	1215	Video	12.766628

In [29]:
```
def replaceNullByZero(col):
    """
    Replace NaN and infinity values by zero
    """
    col_value = col[0]
    col_value = np.nan_to_num(col_value, nan=0, posinf=0, neginf=0)
    return col_value
```

In [32]:
```
df['Engagement Rate']=df[['Engagement Rate']].apply(replaceNullByZero,axis=1)
df.head(100)
```

Out[32]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate
0	03-31-2023 19:55	CSGO	TWITTER	N/A	0	0	Text	0.000000
1	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	0.000000
2	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	0.000000
3	03-31-2023 19:49	CSGO	TWITTER	N/A	0	0	Text	0.000000
4	03-31-2023 19:43	CSGO	TWITTER	Community Engagement	9517	1215	Video	12.766628
...	...	...	...	...	...	...	...	...
95	03-30-2023 11:03	General	INSTAGRAM	N/A	0	0	Text	0.000000
96	03-30-2023 11:03	General	INSTAGRAM	Evil Exhibited	7028	100	Video	1.422880
97	03-30-2023 11:00	DOTA2	TWITTER	N/A	1451	47	Text	3.239145
98	03-30-2023 11:00	DOTA2	TWITTER	N/A	0	0	Video	0.000000
99	03-30-2023 11:00	General	FBPAGE	Evil Exhibited	9273	306	Video	3.299903

100 rows × 8 columns

In [30]:
```
df['Engagement Rate'].describe()
```

Out[30]:
```
count     3479.000000
mean        31.833090
std       1593.821839
min          0.000000
25%          0.000000
50%          2.041949
75%          5.004670
max      94000.000000
Name: Engagement Rate, dtype: float64
```

From the table above, we can see that a typical engagement rate is 31.8% across all social media platforms.

In [31]:
```
df.groupby('Account Type')['Engagement Rate'].describe()
```

Out[31]:

Account Type	count	mean	std	min	25%	50%	75%	max
FBPAGE	585.0	175.564847	3886.331657	0.0	0.000000	8.586312	17.680797	94000.000000
INSTAGRAM	588.0	0.666769	1.567378	0.0	0.000000	0.000000	0.958168	13.273076
LINKEDIN_COMPANY	22.0	1.795018	0.915007	0.0	1.323927	1.689915	2.335601	3.960691
TIKTOK_BUSINESS	113.0	5.316753	2.971130	0.0	3.218302	4.487179	6.951872	16.835700
TWITTER	1951.0	3.248994	3.713681	0.0	0.877716	2.248996	4.496582	60.000000
YOUTUBE	220.0	3.048881	4.102267	0.0	0.000000	2.569124	4.335485	37.931034

From the table above, the mean engagement rate for facebook is questionable since the percentage is greater than 100, which means that for some of the posts, the engagements are actually greater than impressions.

As for other social media platforms, Tiktok has the highest engagement rate around 5.3% and Instagram has the lowest engagement rate of 0.67%.

We should remove the outliers to have a better picture of a typical engagement rate from Facebook.

In [36]:
```
comparison_result = df['Total Engagements'] > df['Total Impressions']
values_greater = df[comparison_result]

# Print the values
values_greater
```

Out[36]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate
837	03-10-2023 12:37	General	FBPAGE	N/A	1	940	Photo	94000.000000
3216	01-19-2023 06:02	General	FBPAGE	N/A	5	44	Photo	880.000000
3457	01-05-2023 11:55	General	FBPAGE	N/A	300	3776	Photo	1258.666667

There are only 3 posts that has more engagements than impressions, we should remove the outliers and check the engagement rate again.

In [37]:
```
# Use boolean indexing to filter out the rows where the condition is True
df_filtered = df[~comparison_result]

df_filtered.groupby('Account Type')['Engagement Rate'].describe()
```

Out[37]:

Account Type	count	mean	std	min	25%	50%	75%	max
FBPAGE	582.0	11.283108	12.151957	0.0	0.000000	8.534606	17.436023	100.000000
INSTAGRAM	588.0	0.666769	1.567378	0.0	0.000000	0.000000	0.958168	13.273076
LINKEDIN_COMPANY	22.0	1.795018	0.915007	0.0	1.323927	1.689915	2.335601	3.960691
TIKTOK_BUSINESS	113.0	5.316753	2.971130	0.0	3.218302	4.487179	6.951872	16.835700
TWITTER	1951.0	3.248994	3.713681	0.0	0.877716	2.248996	4.496582	60.000000
YOUTUBE	220.0	3.048881	4.102267	0.0	0.000000	2.569124	4.335485	37.931034

In [116]:
```
df_filtered['Engagement Rate'].describe()
```

Out[116]:
```
count    3476.000000
mean        4.202720
std         6.736290
min         0.000000
25%         0.000000
50%         2.039731
75%         5.000000
max       100.000000
Name: Engagement Rate, dtype: float64
```

After removing the outliers, we have a clearer picture of the engagement rate from Facebook, 11.3%, which is the highest rate among across all platforms.

Takeaway:

- Since the highest mean engagement rate we are seeing is 11.3% from Facebook, after removing the outliers, it is unlikely we will see a 15% mean enaggement rate across all platforms.

2. Does day of the week and time of posting affect engagement rates

Let's convert `Published Date` to to_datetime object. Day of the week:

- Monday: 0
- Tuesday: 1
- Wednesday: 2
- Thursday: 3
- Friday: 4
- Saturday: 5
- Sunday: 6

In [38]:
```
df['Published Date'] = pd.to_datetime(df['Published Date'])
#day of the week
df['Day of the week'] = df['Published Date'].apply(lambda x: x.weekday())
#time
df['time']=df['Published Date'].apply(lambda x : x.strftime("%H:%M"))
df['time'] = pd.to_datetime(df['time'])

df.head()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_3468\12248604636.py:6: UserWarning:

Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
```

Out[38]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate	Day of the week	time
0	2023-03-31 19:55:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	2023-05-28 19:55:00
1	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	2023-05-28 19:49:00
2	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	2023-05-28 19:49:00
3	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	2023-05-28 19:49:00
4	2023-03-31 19:43:00	CSGO	TWITTER	Community Engagement	9517	1215	Video	12.766628	4	2023-05-28 19:43:00

2-a Day of the Week

In [42]:
```
#removing outliers first
df_filtered = df[~comparison_result]
df_filtered.groupby('Day of the week')['Engagement Rate'].describe()
```

Out[42]:

Day of the week	count	mean	std	min	25%	50%	75%	max
0	234.0	3.590710	5.494242	0.0	0.479198	1.914950	4.258937	37.931034
1	561.0	5.055946	7.629040	0.0	0.000000	2.456992	5.965257	56.480028
2	575.0	3.322831	5.462101	0.0	0.000000	1.717126	4.311582	58.858403
3	704.0	4.915714	8.349649	0.0	0.000000	2.304830	5.450978	100.000000
4	707.0	4.195042	5.896210	0.0	0.505443	2.251704	5.099691	44.713813
5	319.0	3.068647	4.540120	0.0	0.000000	1.409869	4.006944	31.076575
6	376.0	4.297761	7.139118	0.0	0.000000	1.741883	4.746114	55.765629

In [43]:
```
# Calculate the mean engagement rate per day of the week
mean_engagement = df_filtered.groupby('Day of the week')['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Day of the week",
             y="Engagement Rate",
             width=500,
             height=300)

fig.update_layout(
    xaxis_title="Day of the Week",
    yaxis_title="Mean Engagement Rate",
    margin={"r": 0, "t": 0, "b": 0, "l": 50})

fig.show()
```

Tuesday has the highest average engagement rate and Thursday has the highest engagement rate of 100%.

2-b Time

We can create a variable that put times into different time interval and investigate it further.

In [179]:
```
df_filtered['time'] = df_filtered['Published Date'].apply(lambda x: x.strftime("%H:%M"))

def categorize_time(t):
    """
    Categorize time into 8 intervals of 3 hours apart
    """
    hour = int(t.split(':')[0])
    interval = hour // 3  # Divide the hour by 3 to get the interval number
    interval_start = interval * 3  # Calculate the start hour of the interval
    interval_end = interval_start + 3  # Calculate the end hour of the interval
    interval_label = f"{interval_start:02d}-{interval_end:02d}"  # Format the label with leading zeros

    return interval_label

df_filtered['Time of day'] = df_filtered['time'].apply(lambda x: categorize_time(x))
```

In [180]:
```
df_filtered.groupby('Time of day')['Engagement Rate'].describe()
```

Out[180]:

Time of day	count	mean	std	min	25%	50%	75%	max
00-03	2.0	0.420078	0.594080	0.0	0.210039	0.420078	0.630117	0.840156
03-06	46.0	8.201175	13.796341	0.0	1.861152	3.457819	9.278722	56.480028
06-09	377.0	5.694548	8.958138	0.0	0.000000	3.177005	7.513330	100.000000
09-12	1020.0	3.965005	5.933866	0.0	0.000000	2.056412	4.936015	44.713813
12-15	999.0	3.839246	6.224821	0.0	0.000000	1.844854	4.575171	57.889590
15-18	815.0	4.237278	6.565623	0.0	0.486454	2.033454	4.818436	60.000000
18-21	183.0	3.718742	6.064221	0.0	0.000000	1.749664	4.936422	51.560997
21-24	36.0	2.182405	4.528815	0.0	0.000000	0.386954	1.824023	21.649485

In [198]:
```
df_filtered.groupby('Time of day')['Engagement Rate'].mean().reset_index()
```

Out[198]:

	Time of day	Engagement Rate
0	00-03	0.420078
1	03-06	8.201175
2	06-09	5.694548
3	09-12	3.965005
4	12-15	3.839246
5	15-18	4.237278
6	18-21	3.718742
7	21-24	2.182405

In [201]:
```
df_filtered.groupby(['DayOfWeek','TimeOfDay'])['EngagementRate'].mean().reset_index()
```

Out[201]:

	DayOfWeek	TimeOfDay	EngagementRate
0	0	00-03	NaN
1	0	03-06	NaN
2	0	06-09	3.234250
3	0	09-12	4.053950
4	0	12-15	3.948494
5	0	15-18	2.882218
6	0	18-21	2.432243
7	0	21-24	1.077597
8	1	00-03	0.840156
9	1	03-06	38.029915
10	1	06-09	4.342660
11	1	09-12	4.169044
12	1	12-15	4.731023
13	1	15-18	6.606080
14	1	18-21	3.669406
15	1	21-24	1.601376
16	2	00-03	NaN
17	2	03-06	3.846320
18	2	06-09	4.213110
19	2	09-12	2.728241
20	2	12-15	2.837880
21	2	15-18	3.476654
22	2	18-21	2.296104
23	2	21-24	4.023651
24	3	00-03	NaN
25	3	03-06	2.432479
26	3	06-09	8.933079
27	3	09-12	4.833889
28	3	12-15	4.216318
29	3	15-18	4.581347
30	3	18-21	2.769139
31	3	21-24	21.649485
32	4	00-03	0.000000
33	4	03-06	5.871459
34	4	06-09	5.848723
35	4	09-12	4.491755
36	4	12-15	3.615166
37	4	15-18	3.391516
38	4	18-21	5.913853
39	4	21-24	3.112155
40	5	00-03	NaN
41	5	03-06	4.405553
42	5	06-09	5.324937
43	5	09-12	3.861875
44	5	12-15	2.638571
45	5	15-18	1.534563
46	5	18-21	1.145395
47	5	21-24	0.000000
48	6	00-03	NaN
49	6	03-06	NaN
50	6	06-09	5.508482
51	6	09-12	4.054507
52	6	12-15	2.931806
53	6	15-18	5.185564
54	6	18-21	4.450237
55	6	21-24	1.518971

2-c ANOVA test

In [186]:
```
# Create a dictionary to map the current column names to the new column names
column_mapping = {
    'Engagement Rate': 'EngagementRate',
    'Day of the week': 'DayOfWeek',
    'Time of day': 'TimeOfDay'
}

# Use the rename() method to rename the columns
df_filtered = df_filtered.rename(columns=column_mapping)
```

In [193]:
```
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Fit the ANOVA model
model = ols('EngagementRate ~ C(DayOfWeek) + C(TimeOfDay)', data=df_filtered).fit()

# Perform the ANOVA test
anova_table = sm.stats.anova_lm(model, typ=2)

# Print the ANOVA table
print(anova_table)
```

```
                     sum_sq      df         F        PR(>F)
C(DayOfWeek)    1909.640818     6.0  7.164534  1.308818e-07
C(TimeOfDay)    2180.301366     7.0  7.011421  2.516440e-08
Residual      153794.046816  3462.0       NaN           NaN
```

From the ANOVA test, we can see that both `Day of the week` and `Time of day` are significant since the p-value < 0. We can conclude that there are a significant different and `Day of the week` and `Time of day` affect the engagement rate.

Takeaway:

- We should focus on the effects of day of the week and time in respect to increasing company's engagement rate on social media platforms.
- In particular, posting on Tuesday has the highest engagement rate.
- And posting from 3-9am also has the highest engagement rate
- In fact, from our data, posting in between 3-6am on Tuesday has the highest engagement rate of 38%.

3. How are our game titles doing in terms of social performance? Is there a specific game we should focus more on or less?

In [91]:
```
df.head()
```

Out[91]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate	Day of the week	time	Time of day
0	2023-03-31 19:55:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	19:55	evening
1	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	19:49	evening
2	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	19:49	evening
3	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.000000	4	19:49	evening
4	2023-03-31 19:43:00	CSGO	TWITTER	Community Engagement	9517	1215	Video	12.766628	4	19:43	evening

In [90]:
```
df['Account'].unique()
```

Out[90]:
```
array(['CSGO', 'General ', 'DOTA2', 'Content Creators', 'Valorant',
       'General'], dtype=object)
```

We should remove the white spaces from 'General' game title.

In [97]:
```
df['Account'] = df['Account'].str.strip()
df['Account'].unique()
```

Out[97]:
```
array(['CSGO', 'General', 'DOTA2', 'Content Creators', 'Valorant'],
      dtype=object)
```

In [203]:
```
df_filtered = df[~comparison_result]
df_filtered.groupby('Account')['Engagement Rate'].describe()
```

Out[203]:

Account	count	mean	std	min	25%	50%	75%	max
CSGO	270.0	3.030547	3.868445	0.0	0.000000	1.759788	3.452529	17.192041
Content Creators	53.0	4.075632	6.827298	0.0	0.000000	2.478134	4.460967	37.931034
DOTA2	803.0	4.298295	3.929859	0.0	1.528428	3.540813	5.921992	25.389466
General	2290.0	4.341212	7.752974	0.0	0.000000	1.633532	4.502080	100.000000
Valorant	60.0	3.024861	3.581828	0.0	0.000000	2.092041	5.152596	14.285714

In [108]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby('Account')['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Account",
             y="Engagement Rate",
             width=500,
             height=300)

fig.update_layout(
    xaxis_title="Game Title",
    yaxis_title="Mean Engagement Rate",
    margin={"r": 0, "t": 0, "b": 0, "l": 50}
)

fig.show()
```

General, which I assume will be the general account of Evil Geniuses, has the highest engagement rate.

Let's look more into details in terms of different media types.

In [113]:
```
df_filtered.groupby(['Account','Media Type'])['Engagement Rate'].describe()
```

Out[113]:

Account	Media Type	count	mean	std	min	25%	50%	75%	max
CSGO	Link	20.0	0.117280	0.362035	0.000000	0.000000	0.000000	0.000000	1.257862
	Photo	139.0	2.202905	1.546827	0.000000	1.284036	1.830076	2.765259	8.193645
	Text	62.0	1.502418	2.010022	0.000000	0.000000	0.000000	3.315473	7.209063
	Video	49.0	8.500984	5.719709	0.000000	2.380653	10.308642	12.790380	17.192041
Content Creators	Link	3.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Text	12.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Video	38.0	5.684435	7.491092	0.000000	2.345917	3.443694	5.827434	37.931034
DOTA2	Link	9.0	5.613076	6.100481	0.809717	1.944895	2.118003	5.952381	16.873449
	Mixed	2.0	16.732484	2.347624	15.072464	15.902474	16.732484	17.562495	18.392505
	Photo	331.0	4.785465	3.369700	0.000000	2.420792	4.434345	6.436879	25.389466
	Text	343.0	3.007063	3.103658	0.000000	0.805445	2.579769	4.461028	25.000000
	Video	118.0	6.374036	5.514204	0.000000	1.402960	5.477240	9.896970	23.149337
General	Album	4.0	10.000000	20.000000	0.000000	0.000000	0.000000	10.000000	40.000000
	Carousel	9.0	3.784294	2.389164	1.226810	2.542834	3.032566	3.902148	9.272955
	Link	60.0	1.681433	7.805911	0.000000	0.000000	0.061524	0.565989	60.000000
	Mixed	3.0	6.795606	0.393271	6.533617	6.569498	6.605380	6.926601	7.247822
	Photo	1017.0	5.365924	9.672351	0.000000	0.000000	1.254438	5.638629	100.000000
	Text	478.0	2.329303	4.636545	0.000000	0.000000	0.000000	2.482430	37.657809
	Video	719.0	4.416541	5.764871	0.000000	1.384860	2.941176	4.981889	44.993383
Valorant	Link	2.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Text	15.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Video	43.0	4.220736	3.585349	0.000000	0.000000	3.846154	6.886059	14.285714

In [107]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby(['Account','Media Type'])['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Media Type",
             y="Engagement Rate",
             facet_col="Account",
             width=1000,
             height=300)

for axis in fig.select_yaxes():
    axis.title = None

fig.update_layout(
    yaxis_title="Mean Engagement Rate"
)

fig.show()
```

In [118]:
```
fig = px.bar(mean_engagement,
             x="Account",
             y="Engagement Rate",
             color="Media Type",
             width=600,
             height=400)

fig.update_layout(
    yaxis_title="Mean Engagement Rate"
)

fig.show()
```

In [205]:
```
df_filtered.groupby(['Account','Account Type'])[['Total Engagements', 'Total Impressions','Engagement Rate']].mean().reset_index()
```

Out[205]:

	Account	Account Type	Total Engagements	Total Impressions	Engagement Rate
0	CSGO	TWITTER	341.985185	8570.066667	3.030547
1	Content Creators	YOUTUBE	26.169811	852.566038	4.075632
2	DOTA2	TWITTER	150.174843	2252.592453	4.310028
3	DOTA2	YOUTUBE	575.125000	8564.375000	3.132272
4	General	FBPAGE	4420.144330	24867.426117	11.283108
5	General	INSTAGRAM	143.005102	6078.331633	0.666769
6	General	LINKEDIN_COMPANY	38.590909	1945.500000	1.795018
7	General	TIKTOK_BUSINESS	166.946903	3045.654867	5.316753
8	General	TWITTER	443.379233	17376.825056	2.363508
9	General	YOUTUBE	34.434343	1027.505051	2.507024
10	Valorant	YOUTUBE	15.516667	382.983333	3.024861

In [117]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby(['Account', 'Account Type'])['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Account Type",
             y="Engagement Rate",
             facet_col="Account",
             width=1000,
             height=400)

for axis in fig.select_yaxes():
    axis.title = None

fig.update_layout(
    yaxis_title="Mean Engagement Rate"
)

fig.show()
```

Content creators and Valorant generally post videos on Youtube platform only. The general Evil Geniuses account posts all types of social media content across all socia media platform.

In [124]:
```
df_filtered.groupby(['Account'])[['Total Engagements', 'Total Impressions', 'Engagement Rate']].mean().reset_index()
```

Out[124]:

	Account	Total Engagements	Total Impressions	Engagement Rate
0	CSGO	341.985185	8570.066667	3.030547
1	Content Creators	26.169811	852.566038	4.075632
2	DOTA2	154.408468	2315.474471	4.298295
3	General	1341.732751	14817.227511	4.341212
4	Valorant	15.516667	382.983333	3.024861

The engagement rates are similar across the game titles. The general Evil Geniuses account has more enagements and impressions than other titles, followed by CSGO and DOTA2.

Takeaway:

General, DOTA2 and CSGO are doing fairly well. If we want to improve the overall performance of the game titles, we should focus on improving the impressions of Valorant and posts from Content Creators.

In addition, having more detailed information on each game title would give us a better perspectives of the overall social media activities. For example, if we can access data on each game title of Evil Geniuses in combination with daily increases of new users on each game plateform, we can calculate the conversion rate and investigate further.

4. What media type performs the best?

In [127]:
```
df_filtered.groupby(['Media Type','Account Type'])[['Total Engagements', 'Total Impressions', 'Engagement Rate']
```

Out[127]:

	Media Type	Account Type	Total Engagements	Total Impressions	Engagement Rate
0	Album	FBPAGE	2.000000	5.000000	10.000000
1	Carousel	INSTAGRAM	726.777778	17854.333333	3.784294
2	Link	FBPAGE	3.714286	458.285714	0.057891
3	Link	INSTAGRAM	0.000000	0.000000	0.000000
4	Link	LINKEDIN_COMPANY	64.666667	2584.000000	2.310642
5	Link	TWITTER	28.323944	3074.605634	2.056435
6	Link	YOUTUBE	0.000000	0.000000	0.000000
7	Mixed	TWITTER	2732.800000	36996.800000	10.770357
8	Photo	FBPAGE	5240.313043	28593.101449	13.647598
9	Photo	INSTAGRAM	137.595745	4753.035461	0.459752
10	Photo	LINKEDIN_COMPANY	41.538462	1960.923077	1.934351
11	Photo	TWITTER	372.761511	14950.670602	2.932845
12	Text	FBPAGE	1526.082707	9051.872180	3.927818
13	Text	INSTAGRAM	0.000000	0.000000	0.000000
14	Text	LINKEDIN_COMPANY	11.000000	1660.750000	0.613156
15	Text	TWITTER	67.973312	2975.919937	2.689368
16	Text	YOUTUBE	0.000000	0.000000	0.000000
17	Video	FBPAGE	6530.081395	39510.313953	15.059735
18	Video	INSTAGRAM	193.720000	10365.070000	1.141756
19	Video	LINKEDIN_COMPANY	35.500000	1457.000000	2.479645
20	Video	TIKTOK_BUSINESS	166.946903	3045.654867	5.316753
21	Video	TWITTER	587.867008	11607.253197	4.965940
22	Video	YOUTUBE	59.017143	1362.302857	3.832879

If we take into account of the social media platforms, we can see that Album is only posted on Facebook, Carousel is only posted on instagram, and Mixed is only posted on Twitter.

Since album is essentially a college of photos, we can put album into the same category as photo.

In [132]:
```
# Convert 'album' to 'photo' in the 'media type' column
df_filtered['Media Type'] = df_filtered['Media Type'].replace('Album', 'Photo')
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_3448\3090891966.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
```

In [133]:
```
df_filtered.groupby(['Media Type'])[['Total Engagements', 'Total Impressions','Engagement Rate']].mean().reset_in
```

Out[133]:

	Media Type	Total Engagements	Total Impressions	Engagement Rate
0	Carousel	726.777778	17854.333333	3.784294
1	Link	24.010638	2473.031915	1.635631
2	Mixed	2732.800000	36996.800000	10.770357
3	Photo	1450.695506	16025.289068	4.954619
4	Text	270.672527	3413.409890	2.459318
5	Video	888.780765	10956.373320	4.903493

In [134]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby(['Media Type'])['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Media Type",
             y="Engagement Rate",
             width=600,
             height=400)

fig.update_layout(
    yaxis_title="Mean Engagement Rate"
)

fig.show()
```

In [135]:
```
df_filtered.groupby(['Media Type','Account Type'])[['Total Engagements', 'Total Impressions', 'Engagement Rate']
```

Out[135]:

	Media Type	Account Type	Total Engagements	Total Impressions	Engagement Rate
0	Carousel	INSTAGRAM	726.777778	17854.333333	3.784294
1	Link	FBPAGE	3.714286	458.285714	0.057891
2	Link	INSTAGRAM	0.000000	0.000000	0.000000
3	Link	LINKEDIN_COMPANY	64.666667	2584.000000	2.310642
4	Link	TWITTER	28.323944	3074.605634	2.056435
5	Link	YOUTUBE	0.000000	0.000000	0.000000
6	Mixed	TWITTER	2732.800000	36996.800000	10.770357
7	Photo	FBPAGE	5180.275072	28265.444126	13.605791
8	Photo	INSTAGRAM	137.595745	4753.035461	0.459752
9	Photo	LINKEDIN_COMPANY	41.538462	1960.923077	1.934351
10	Photo	TWITTER	372.761511	14950.670602	2.932845
11	Text	FBPAGE	1526.082707	9051.872180	3.927818
12	Text	INSTAGRAM	0.000000	0.000000	0.000000
13	Text	LINKEDIN_COMPANY	11.000000	1660.750000	0.613156
14	Text	TWITTER	67.973312	2975.919937	2.689368
15	Text	YOUTUBE	0.000000	0.000000	0.000000
16	Video	FBPAGE	6530.081395	39510.313953	15.059735
17	Video	INSTAGRAM	193.720000	10365.070000	1.141756
18	Video	LINKEDIN_COMPANY	35.500000	1457.000000	2.479645
19	Video	TIKTOK_BUSINESS	166.946903	3045.654867	5.316753
20	Video	TWITTER	587.867008	11607.253197	4.965940
21	Video	YOUTUBE	59.017143	1362.302857	3.832879

In [138]:
```
df_filtered.groupby(['Media Type'])['Engagement Rate'].describe().reset_index()
```

Out[138]:

	Media Type	count	mean	std	min	25%	50%	75%	max
0	Carousel	9.0	3.784294	2.389164	1.226810	2.542834	3.032566	3.902148	9.272955
1	Link	94.0	1.635631	6.637472	0.000000	0.000000	0.000000	0.810285	60.000000
2	Mixed	5.0	10.770357	5.574731	6.533617	6.605380	7.247822	15.072464	18.392505
3	Photo	1491.0	4.954619	8.260214	0.000000	0.000000	1.949306	5.604760	100.000000
4	Text	910.0	2.459318	3.942132	0.000000	0.000000	1.122761	3.476848	37.657809
5	Video	967.0	4.903493	5.818951	0.000000	1.423765	3.190552	6.241128	44.993383

If we use Engagement Rate as the metrics, Mixed media type performs the best among all media types, and Video and Photo has similar performance. In terms of total impressions, mixed media type still has the highest views, followed by Carousel and Photo.

Takeaway:

- Photo, Video, and Text are the most common types of posts and ads on social media platform.
- Mixed media type is only posted on Twitter, so we can infer that Twitter has a high engagement rate. Thus, posting mixed media type on Twitter is an efficient way to boost the company's social media present.
- Carousel is only posted on Instagram. Similarly, posting Carousel media type is more efficient than other media type posts on Instagram.
- Link on Youtube and Text on Instagram have little effect on social media performance.
- Link works best on LinkedIn, although the engagement rate is still fairly low.
- Photo, Video, Text works best on Facebook and has an engagement rate better than any other social media platform.
- Overall, if the company intended on increasing its engagement on social media, Facebook should be the go-to platform.

5. What is our best performing campaign?

In [142]:
```
df_filtered.groupby(['Campaign Name'])['Engagement Rate'].describe().reset_index()
```

Out[142]:

	Campaign Name	count	mean	std	min	25%	50%	75%	max
0	Community Engagement	1411.0	5.634547	7.575413	0.0	1.144285	2.882206	6.627286	58.858403
1	Evergreen	163.0	3.467474	2.487111	0.0	1.661541	3.090909	4.413168	13.649289
2	Evil Exhibited	420.0	2.681813	2.548381	0.0	0.854577	2.064265	3.856838	15.988244
3	N/A	1482.0	3.351382	6.777150	0.0	0.000000	0.000000	4.195538	100.000000

In [145]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby(['Campaign Name'])['Engagement Rate'].mean().reset_index()

fig = px.bar(mean_engagement,
             x="Campaign Name",
             y="Engagement Rate",
             width=600,
             height=400)

fig.update_layout(
    yaxis_title="Mean Engagement Rate"
)

fig.show()
```

The best performing campaign is Community Engagement, followed by Evergreen.

In [146]:
```
df_filtered.groupby(['Campaign Name','Account Type'])['Engagement Rate'].describe().reset_index()
```

Out[146]:

	Campaign Name	Account Type	count	mean	std	min	25%	50%	75%	max
0	Community Engagement	FBPAGE	301.0	14.505180	11.058025	0.000000	6.284133	12.971957	20.639778	58.858403
1	Community Engagement	INSTAGRAM	159.0	1.208653	2.484503	0.000000	0.000000	0.000000	1.366499	13.273076
2	Community Engagement	LINKEDIN_COMPANY	18.0	1.860454	0.779330	0.686591	1.398350	1.689915	2.311451	3.960691
3	Community Engagement	TIKTOK_BUSINESS	13.0	6.422724	2.324590	3.973510	4.683270	5.732484	7.241911	11.067708
4	Community Engagement	TWITTER	899.0	3.468913	3.335065	0.000000	1.269614	2.369668	4.625469	34.639889
5	Community Engagement	YOUTUBE	21.0	7.455909	9.598989	0.000000	2.681292	4.334365	5.555556	37.931034
6	Evergreen	FBPAGE	1.0	13.098000	NaN	13.098000	13.098000	13.098000	13.098000	13.098000
7	Evergreen	INSTAGRAM	64.0	1.744471	0.877822	0.656168	1.139780	1.513705	1.968753	5.219921
8	Evergreen	TIKTOK_BUSINESS	63.0	4.832811	2.523657	1.208459	3.067017	4.237288	6.604817	13.649289
9	Evergreen	TWITTER	1.0	11.770774	NaN	11.770774	11.770774	11.770774	11.770774	11.770774
10	Evergreen	YOUTUBE	34.0	3.653416	1.636273	0.000000	3.097628	3.727058	4.500504	8.333333
11	Evil Exhibited	FBPAGE	34.0	3.513447	2.349649	0.625835	1.778778	3.243064	4.880881	10.298713
12	Evil Exhibited	INSTAGRAM	72.0	0.815349	1.121810	0.000000	0.000000	0.000000	1.425745	7.094595
13	Evil Exhibited	LINKEDIN_COMPANY	1.0	3.213268	NaN	3.213268	3.213268	3.213268	3.213268	3.213268
14	Evil Exhibited	TIKTOK_BUSINESS	22.0	3.770242	1.236089	1.587302	3.054648	3.768259	4.460745	7.189542
15	Evil Exhibited	TWITTER	230.0	2.961700	2.427204	0.000000	1.139953	2.345297	4.032413	15.988244
16	Evil Exhibited	YOUTUBE	61.0	2.964750	3.523174	0.000000	0.000000	2.013423	5.151915	14.285714
17	N/A	FBPAGE	246.0	8.407131	13.048530	0.000000	0.000000	0.000000	15.039087	100.000000
18	N/A	INSTAGRAM	293.0	0.100795	0.620454	0.000000	0.000000	0.000000	0.000000	7.207207
19	N/A	LINKEDIN_COMPANY	3.0	0.929658	1.208698	0.000000	0.246479	0.492958	1.394486	2.296015
20	N/A	TIKTOK_BUSINESS	15.0	8.659015	4.192338	0.000000	5.803398	8.058252	11.481317	16.835700
21	N/A	TWITTER	821.0	3.078287	4.333225	0.000000	0.000000	1.862197	4.530201	60.000000
22	N/A	YOUTUBE	104.0	2.010710	2.151343	0.000000	0.000000	1.716897	3.315322	7.629108

The Community Engagement campaign is mostly advertised on Twitter and Facebook. The Evergreen campaign is mostly advertised on Instagram and Tiktok. Evil Exhibited Campaign is mostly advertised Twitter, Instagram and Youtube.

Takeaway:

- Community Engagement is the most successful campaign with a mean engagement rate of 5.6%, in which the campaign has the highest engagement rate on Facebook even though the campaign is more widely advertised on Twitter.
- Evergreen campaign is most effective on Facebook, with a mean engagement rate of 13%.
- Evil Exhibited campaign is mostly advertised on Twitter, but the engagement rate is higher on Tiktok.

Further exploration

In [209]:
```
df_filtered.groupby(['Account Type'])[['Total Engagements', 'Total Impressions','Engagement Rate']].mean().reset_
```

Out[209]:

	Account Type	Total Engagements	Total Impressions	Engagement Rate
0	FBPAGE	4420.144330	24867.426117	11.283108
1	INSTAGRAM	143.005102	6078.331633	0.666769
2	LINKEDIN_COMPANY	38.590909	1945.500000	1.795018
3	TIKTOK_BUSINESS	166.946903	3045.654867	5.316753
4	TWITTER	309.871348	9995.179908	3.248994
5	YOUTUBE	46.945455	1083.650000	3.048881

In [158]:
```
# Calculate the mean engagement rate
mean_engagement = df_filtered.groupby(['Account Type'])['Engagement Rate'].mean().reset_index()

# Sort the DataFrame by 'Engagement Rate' in descending order
mean_engagement = mean_engagement.sort_values('Engagement Rate', ascending=False)

# Plot the bar graph
fig = px.bar(mean_engagement, x="Account Type", y="Engagement Rate", width=600, height=400)
fig.update_layout(yaxis_title="Mean Engagement Rate")
fig.show()
```

In addition to the qeustions above, it is also beneficial to look into the social media performance on each social media platform. If we use engagement rate as a metric, Facebook has the highest mean engagement rate, followed by Tiktok, Twitter and Youtube. At the same time, Facebook engagement rate also has a large variance in comparison to other social media types.

In [161]:
```
# Filter the data where 'Engagement Rate' is 0 for each 'Account Type' group
filtered_data = df[df['Engagement Rate'] == 0]

filtered_data
```

Out[161]:

	Published Date	Account	Account Type	Campaign Name	Total Impressions	Total Engagements	Media Type	Engagement Rate	Day of the week	Time	Time of day
0	2023-03-31 19:55:00	CSGO	TWITTER	N/A	0	0	Text	0.0	4	19:55	evening
1	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.0	4	19:49	evening
2	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.0	4	19:49	evening
3	2023-03-31 19:49:00	CSGO	TWITTER	N/A	0	0	Text	0.0	4	19:49	evening
4	2023-03-31 19:42:00	CSGO	TWITTER	N/A	0	0	Photo	0.0	4	19:42	evening
...	...	...	...	...	...	...	...	...	...	...	...
3473	2023-01-03 10:35:00	CSGO	TWITTER	N/A	0	0	Link	0.0	1	10:35	morning
3474	2023-01-03 10:35:00	CSGO	TWITTER	N/A	0	0	Link	0.0	1	10:35	morning
3475	2023-01-03 10:34:00	CSGO	TWITTER	N/A	0	0	Link	0.0	1	10:34	morning
3476	2023-01-03 10:34:00	CSGO	TWITTER	N/A	0	0	Link	0.0	1	10:34	morning
3478	2023-01-01 14:59:00	General	FBPAGE	N/A	0	0	Photo	0.0	6	14:59	afternoon

966 rows × 11 columns

In [169]:
```
count_per_account_type = filtered_data.groupby('Account Type').size()
count_per_account_type
```

Out[169]:
```
Account Type
FBPAGE              153
INSTAGRAM           432
LINKEDIN_COMPANY      1
TIKTOK_BUSINESS       1
TWITTER             302
YOUTUBE              77
dtype: int64
```

In [171]:
```
total_rows_per_account_type = df.groupby('Account Type').size()
#calculate the percentage of zero engagement rate on each platforms
percentage_zero_engagement = (count_per_account_type / total_rows_per_account_type) * 100

# Print the percentage of zero engagement rate rows in each account type
percentage_zero_engagement
```

Out[171]:
```
Account Type
FBPAGE              26.153846
INSTAGRAM           73.469388
LINKEDIN_COMPANY     4.545455
TIKTOK_BUSINESS      0.884956
TWITTER             15.479241
YOUTUBE             35.000000
dtype: float64
```

Overall, Youtube has about 35% of chance in getting zero enagement rate, followed by Facebook with 26%.

6. Define out a posting strategy for our social channels based on your discoveries.

Here are some statistics about the demographics in US Social Media Demographics:

- Facebook and YouTube are both prime places for ads due in part to their high-earning user bases.
- 71% of consumers and 65% of marketers plan on using Facebook the most in the next year. YouTube follows behind with 51% of consumers and 35% of marketers planning to use the platform.
- The top social networks among Millennials and Gen Z are Instagram and YouTube, signaling the strength of bold, eye-popping content that oozes with personality.
- LinkedIn's user base is well-educated, making it a hub for in-depth or industry-specific content that might be more niche than what you see on Facebook or Twitter.
- Consumers in the United States spend the most time on TikTok. And some 73% of users feel a deeper connection to brands they interact with on TikTok vs other platforms.

Posting Strategy:

1. Focus on Day of the Week and Time of Posting:

Based on the analysis, Tuesday has the highest average engagement rate, and posting from 3-9am generally has the highest engagement rate. Consider scheduling posts on Tuesday and during the early morning hours (3-9am) to maximize engagement.

1. Pay Attention to Different Media Types:

Mixed media type performs the best in terms of engagement rate, followed by Video and Photo. Focus on creating and sharing mixed media content on Twitter, as it has shown to have a high engagement rate. Utilize Carousel media type specifically for Instagram, as it has a better performance compared to other media types on that platform.

1. Game Titles and Campaigns:

General, DOTA2, and CSGO are performing well in terms of engagement and impressions. Focus on maintaining and improving their performance. Pay special attention to Valorant and Content Creators to improve their overall social media performance by increasing impressions and engagement. The Community Engagement campaign has shown the highest engagement rate. Continue to promote and advertise it on Twitter and Facebook. Evergreen campaign performs well on Facebook, while the Evil Exhibited campaign has higher engagement on Tiktok. Allocate resources accordingly.

1. Facebook as the Go-To Platform:

Facebook has the highest mean engagement rate among all social media platforms analyzed. Consider prioritizing Facebook for overall engagement, as it has a higher chance of achieving good engagement rates.

1. Further Analysis Opportunities:

Explore more detailed information on each game title and daily increases in new users to calculate conversion rates and investigate further opportunities for improvement. Analyze social media performance on each platform individually to gain insights specific to each platform and adapt strategies accordingly. Remember to regularly monitor the performance of company's social media channels, track engagement metrics, and make adjustments to the posting strategy as needed.

In addition, Instagram seems to not be doing so well in terms of engaging with its users, so here are some ideas to explore to increase the overall performance on Instagram:

- Analyze the content and messaging of the company's posts on Instagram. Ensure that they are visually appealing, engaging, and aligned with the target audience's preferences.
- Experiment with different types of content, such as high-quality photos, videos, user-generated content, and behind-the-scenes footage, to see what resonates best with the audience.
- Utilize Instagram's features like Stories, IGTV, and Reels to diversify contents and capture the attention of account's followers.
- Engage with the audience by responding to comments, direct messages, and mentions promptly. Building a strong community and fostering connections can help boost engagement.
- Collaborate with influencers or popular accounts in the gaming industry to increase reach and exposure on the platform.
- Use relevant *hashtags* strategically to increase discoverability and reach on Instagram.

7. What suggestions would you give to the social media team if they want to expand their presence (e.g. if our CSGO youtube channel is doing well should we expand to TikTok)?

Ultimately, we would need more data to make the decisions, such as conversion rates from different platform. But based on the data we have so far, here are some sugestions:

1. Analyze Audience and Platform Fit:
    - Research the demographics and user behavior of the target audience on TikTok. Understand if the demographics align with the brand's target audience for the CSGO YouTube channel.
    - Evaluate if the content format and style that performs well on YouTube would resonate with TikTok users. Consider the differences in content duration, editing style, and engagement patterns on TikTok.
    - Assess if the content from the CS:GO YouTube channel can be repurposed or adapted to suit the TikTok platform.
2. Explore TikTok Potential:
    - Conduct a competitive analysis to see if other CSGO or gaming-related accounts are successful on TikTok. Assess the engagement levels and the types of content that perform well within the CSGO community on TikTok.
    - Experiment with creating TikTok content that aligns with the interests and preferences of the TikTok gaming community.
    - Leverage TikTok's unique features, such as music, effects, challenges, and trends, to engage with the audience and showcase the CSGO content in a creative and entertaining way.
3. Develop a Content Strategy:
    - Identify the key strengths and unique aspects of the CSGO YouTube channel that can be leveraged on TikTok. Determine how to translate those strengths into TikTok-friendly content.
    - Plan and create engaging and shareable content specifically tailored for TikTok's short-form video format.
    - Consider using behind-the-scenes footage, highlights, tips and tricks, funny moments, or challenges related to CSGO to capture the attention of the TikTok audience.
4. Cross-Promotion and Integration:
    - Leverage the existing audience and reach on the CSGO YouTube channel to promote and drive traffic to TikTok content. Cross-promote the TikTok content on YouTube by including links or call-to-actions in the video descriptions or community posts.
    - Collaborate with popular TikTok creators or influencers in the gaming niche to expand the reach and tap into their existing audience.
    - Integrate the branding and messaging consistently across both platforms to create a cohesive presence and strengthen the overall brand identity.
5. Monitor and Optimize:
    - Regularly monitor the performance of the TikTok content. Track engagement metrics, analyze audience feedback, and adapt the content strategy based on the insights gained.
    - Experiment with different content formats, hashtags, and trends on TikTok to understand what resonates best with the audience and adjust the approach accordingly.
    - Continuously learn from the TikTok community, stay updated on trends, and iterate on content strategy to stay relevant and maintain growth.

Expanding to additional social media platfofrms can provide an opportunity to reach a new and potentially broader audience. However, it's important to carefully assess the fit between the brand, content, and the platform. With strategic planning, content adaptation, and audience analysis, we can leverage the presence on Youtube and expand social media presence more effectively.

Thank you.