In [374]:

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

Sales Trend Analysis

In [375]:

```python
df= pd.read_csv("sales_2023-05-01_2024-04-01.csv")
df.head(12)
```

Out[375]:

	month	orders	gross_sales	discounts	returns	net_sales	shipping	duties	additional
0	2023-05	0	0.0	0.00	0.0	0.00	0.0	0.0	
1	2023-06	18	39815.0	0.00	0.0	39815.00	1194.0	0.0	
2	2023-07	35	116930.0	-899.50	-3495.0	112535.50	3582.0	0.0	
3	2023-08	49	242370.0	-13710.15	-56639.9	172019.95	6917.0	0.0	
4	2023-09	39	126258.0	0.00	-49600.0	76658.00	5976.0	0.0	
5	2023-10	26	113335.0	0.00	-11290.0	102045.00	3735.0	0.0	
6	2023-11	54	175505.0	0.00	-4790.0	170715.00	10458.0	0.0	
7	2023-12	46	186989.0	-77.00	-29555.0	157357.00	8217.0	0.0	
8	2024-01	27	74960.0	0.00	-20160.0	54800.00	4233.0	0.0	
9	2024-02	36	73763.0	0.00	-38469.0	35294.00	4233.0	0.0	
10	2024-03	41	98530.0	-1695.00	-7589.0	89246.00	8964.0	0.0	
11	2024-04	35	154209.0	0.00	-8925.0	145284.00	5976.0	0.0	

In [376]:

```
df.info()
```

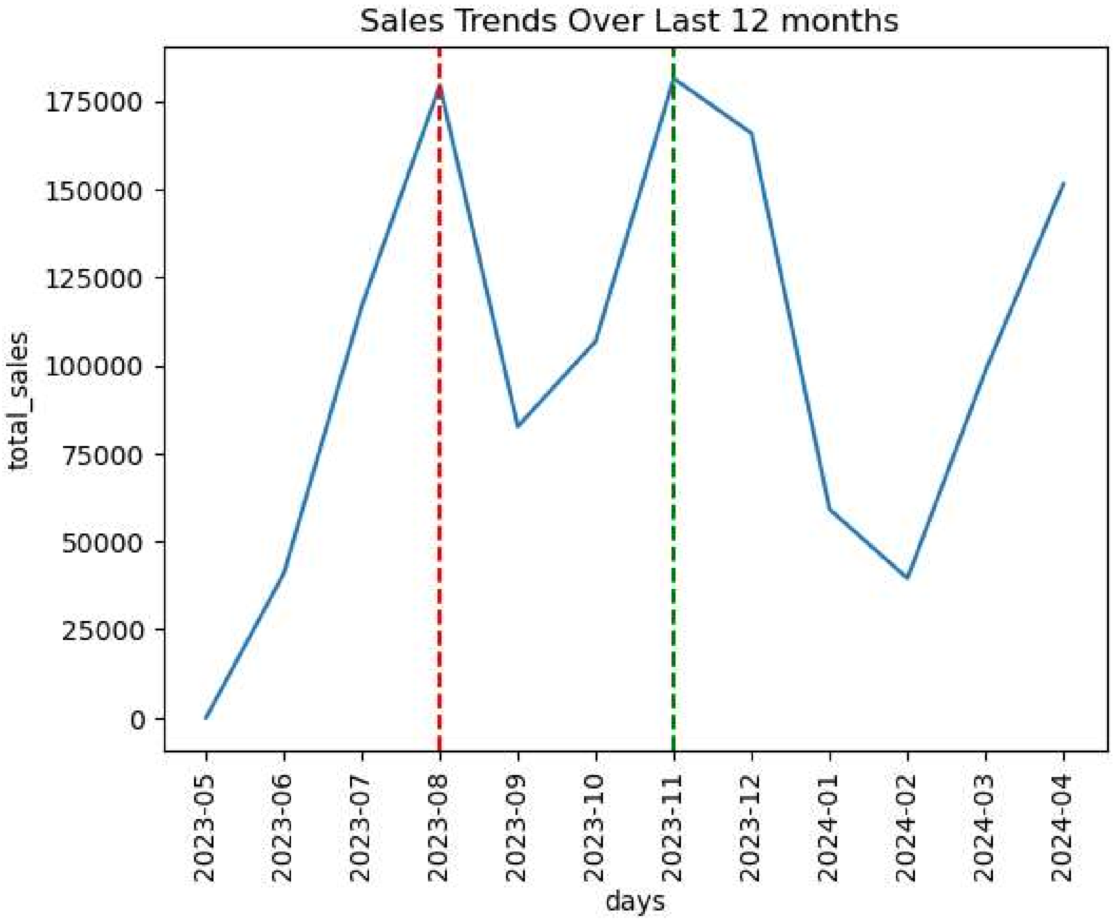
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   month            12 non-null     object 
 1   orders           12 non-null     int64  
 2   gross_sales      12 non-null     float64
 3   discounts        12 non-null     float64
 4   returns          12 non-null     float64
 5   net_sales        12 non-null     float64
 6   shipping         12 non-null     float64
 7   duties           12 non-null     float64
 8   additional_fees  12 non-null     float64
 9   taxes            12 non-null     float64
 10  total_sales      12 non-null     float64
dtypes: float64(9), int64(1), object(1)
memory usage: 1.2+ KB
```

In [377]:

```
plt.plot(df["month"],df["total_sales"])
plt.xticks(rotation=90)
plt.axvline(x=3, color='r', linestyle='--')
plt.axvline(x=6, color='g', linestyle='--')
plt.xlabel("days")
plt.ylabel("total_sales")
plt.title("Sales Trends Over Last 12 months ")
```

Out[377]:

```
Text(0.5, 1.0, 'Sales Trends Over Last 12 months ')
```

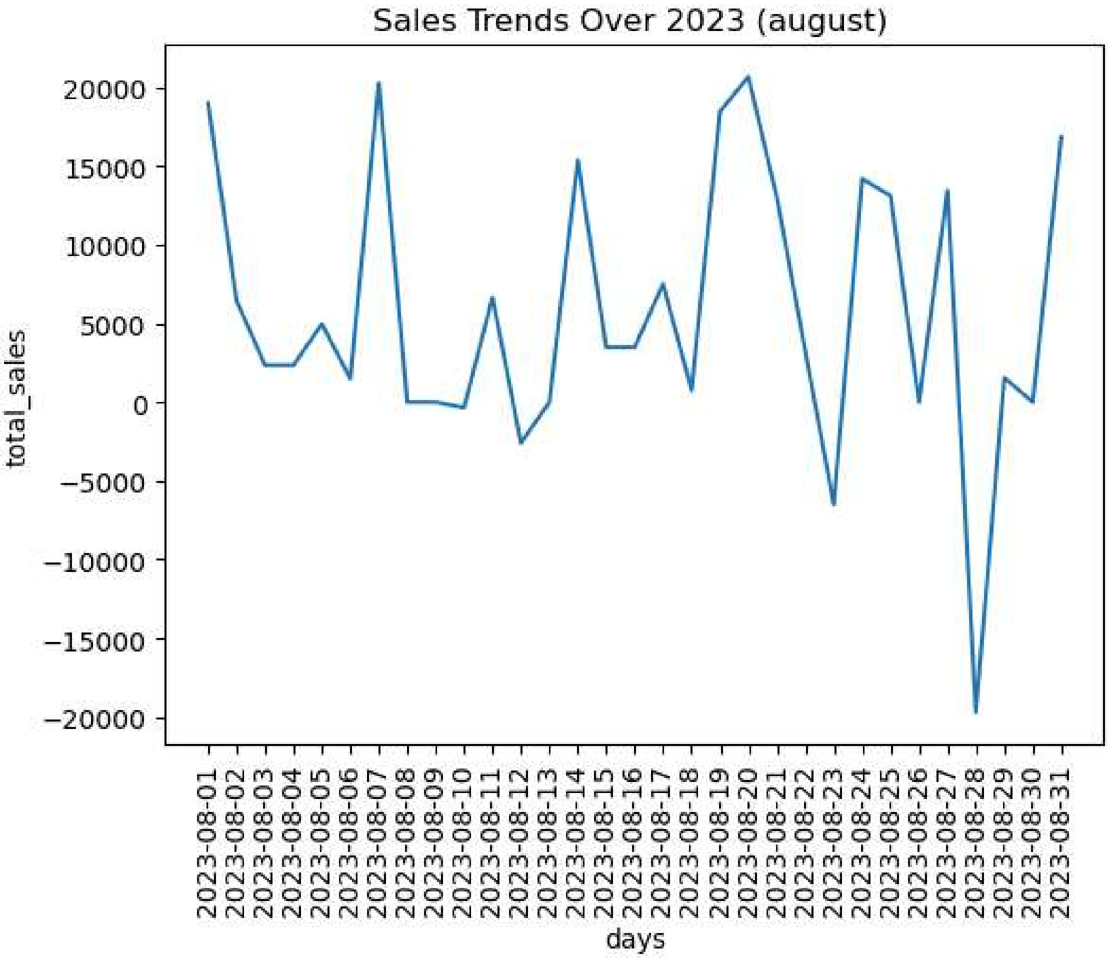


In [378]:

```
df2=pd.read_csv("sales_2023-08-01_2023-08-31.csv")
plt.plot(df2["day"],df2["total_sales"])
plt.xticks(rotation=90)
#plt.axvline(x=3, color='r', linestyle='--')
#plt.axvline(x=6, color='g', linestyle='--')
plt.xlabel("days")
plt.ylabel("total_sales")
plt.title("Sales Trends Over 2023 (august) ")
```

Out[378]:

```
Text(0.5, 1.0, 'Sales Trends Over 2023 (august) ')
```



In [379]:

```
df2.head(30)
```

Out[379]:

	day	orders	gross_sales	discounts	returns	net_sales	shipping	duties	additional_
0	2023-08-01	5	18780.0	0.00	0.0	18780.00	199.0	0.0	
1	2023-08-02	1	7490.0	-1048.60	0.0	6441.40	0.0	0.0	
2	2023-08-03	1	2495.0	-349.30	0.0	2145.70	199.0	0.0	
3	2023-08-04	1	2495.0	-349.30	0.0	2145.70	199.0	0.0	
4	2023-08-05	4	11480.0	-1607.20	-5500.7	4372.10	597.0	0.0	
5	2023-08-06	1	1495.0	-209.30	0.0	1285.70	199.0	0.0	
6	2023-08-07	6	40040.0	-5640.55	-14942.5	19456.95	796.0	0.0	
7	2023-08-08	0	0.0	0.00	0.0	0.00	0.0	0.0	
8	2023-08-09	0	0.0	0.00	0.0	0.00	0.0	0.0	
9	2023-08-10	1	7500.0	-1050.00	-6995.0	-545.00	199.0	0.0	
10	2023-08-11	1	7500.0	-1050.00	0.0	6450.00	199.0	0.0	
11	2023-08-12	0	0.0	0.00	-2595.0	-2595.00	0.0	0.0	
12	2023-08-13	0	0.0	0.00	0.0	0.00	0.0	0.0	
13	2023-08-14	3	17185.0	-2405.90	0.0	14779.10	597.0	0.0	
14	2023-08-15	1	3295.0	0.00	0.0	3295.00	199.0	0.0	
15	2023-08-16	1	3295.0	0.00	0.0	3295.00	199.0	0.0	
16	2023-08-17	2	9540.0	0.00	-2231.7	7308.30	199.0	0.0	
17	2023-08-18	1	550.0	0.00	0.0	550.00	199.0	0.0	
18	2023-08-19	1	18265.0	0.00	0.0	18265.00	199.0	0.0	
19	2023-08-20	4	19865.0	0.00	0.0	19865.00	796.0	0.0	
20	2023-08-21	3	11875.0	0.00	0.0	11875.00	647.0	0.0	
21	2023-08-22	1	2995.0	0.00	0.0	2995.00	249.0	0.0	

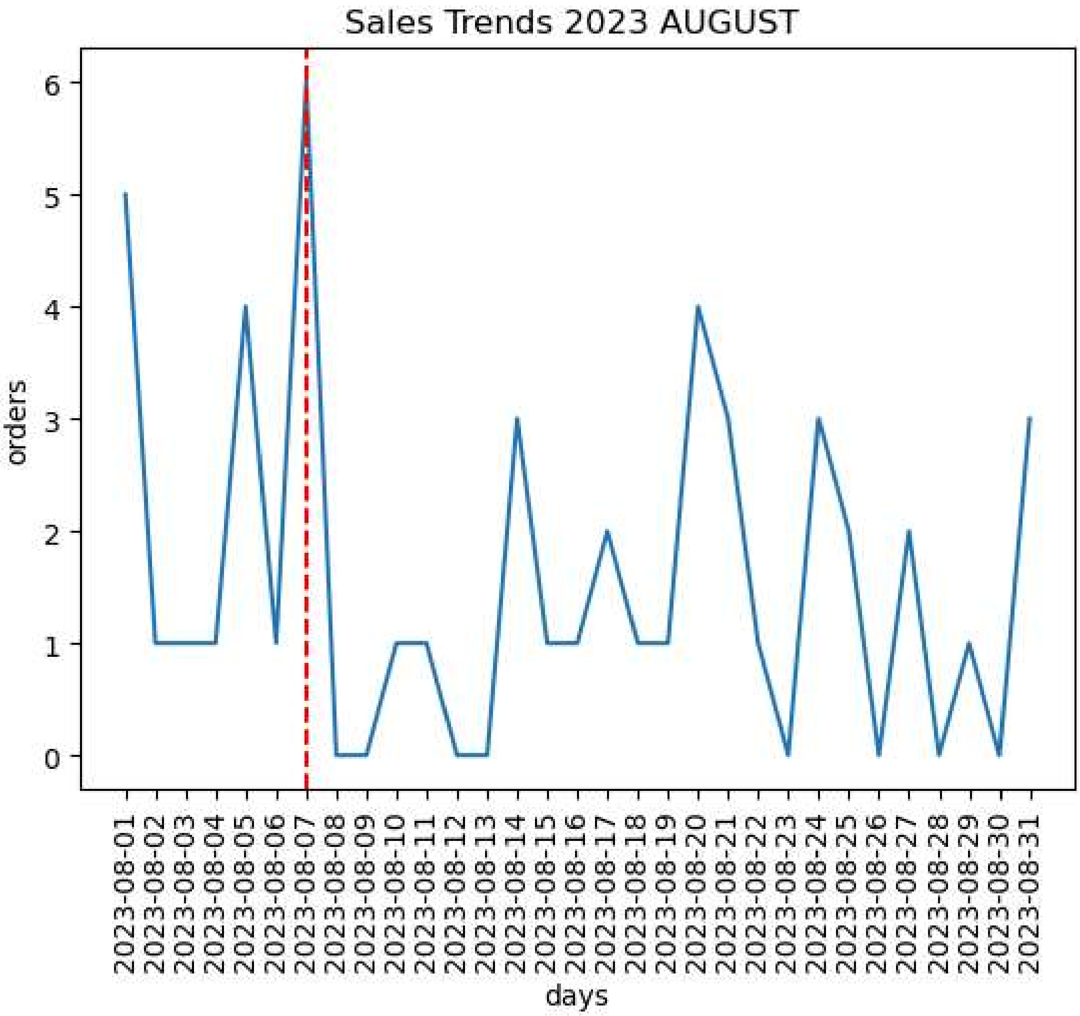
	day	orders	gross_sales	discounts	returns	net_sales	shipping	duties	additional_
22	2023-08-23	0	0.0	0.00	-6290.0	-6290.00	-199.0	0.0	
23	2023-08-24	3	13680.0	0.00	0.0	13680.00	498.0	0.0	
24	2023-08-25	2	11495.0	0.00	0.0	11495.00	249.0	0.0	
25	2023-08-26	0	0.0	0.00	0.0	0.00	0.0	0.0	
26	2023-08-27	2	13180.0	0.00	0.0	13180.00	249.0	0.0	
27	2023-08-28	0	0.0	0.00	-18085.0	-18085.00	-249.0	0.0	
28	2023-08-29	1	1295.0	0.00	0.0	1295.00	249.0	0.0	
29	2023-08-30	0	0.0	0.00	0.0	0.00	0.0	0.0	

In [380]:

```
plt.plot(df2["day"],df2["orders"])
plt.xticks(rotation=90)
plt.axvline(x=6, color='r', linestyle='--')
#plt.axvline(x=6, color='g', linestyle='--')
plt.xlabel("days")
plt.ylabel("orders")
plt.title("Sales Trends 2023 AUGUST ")
```

Out[380]:

```
Text(0.5, 1.0, 'Sales Trends 2023 AUGUST ')
```



Performance Analysis

In [381]:

```
df3= pd.read_csv("sales_2024-01-01_2024-04-30.csv")
df3.head()
```

Out[381]:

	referring_channel	referring_category	fulfillment_status	order_id	purchase_option
0	facebook	social	fulfilled	5792835698989	One-time
1	facebook	social	fulfilled	5698497118509	One-time
2	direct	NaN	fulfilled	5704323432749	One-time
3	facebook	social	fulfilled	5787960901933	One-time
4	facebook	social	fulfilled	5792835698989	One-time

5 rows × 23 columns

In [382]:

```
df4 = pd.read_csv("sale_2024-01-01_2024-04-30.csv")
```

In [383]:

```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 459 entries, 0 to 458
Data columns (total 80 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   day                                 459 non-null    object
 1   adjustment                          425 non-null    object
 2   market_name                         425 non-null    object
 3   customer_cohort_quarter             459 non-null    object
 4   customer_cohort_month               459 non-null    object
 5   customer_cohort_week                459 non-null    object
 6   referring_traffic                   425 non-null    object
 7   referring_category                  376 non-null    object
 8   referring_channel                   425 non-null    object
 9   referring_platform                  425 non-null    object
 10  referrer_url                        336 non-null    object
 11  referrer_source                     425 non-null    object
 12  referrer_path                       336 non-null    object
 13  referrer_name                       317 non-null    object
 14  referrer_host                       336 non-null    object
 15  id_of_staff_who_helped_with_sale    459 non-null    int64
 16  name_of_staff_who_helped_with_sale  0 non-null      float64
 17  staff_name                          10 non-null     object
 18  staff_id                            459 non-null    int64
 19  shipping_postal_code                357 non-null    object
 20  shipping_country                    422 non-null    object
 21  shipping_region                     0 non-null      float64
 22  shipping_city                       422 non-null    object
 23  api_client_title                    425 non-null    object
 24  variant_title                       0 non-null      float64
 25  variant_sku                         255 non-null    float64
 26  variant_id                          459 non-null    int64
 27  product_type                        255 non-null    object
 28  product_price                       459 non-null    float64
 29  product_id                          459 non-null    int64
 30  utm_campaign_term                   0 non-null      float64
 31  utm_campaign_source                 345 non-null    object
 32  utm_campaign_name                   4 non-null      object
 33  utm_campaign_medium                 322 non-null    object
 34  pos_location_name                   0 non-null      float64
 35  utm_campaign_content                341 non-null    object
 36  marketing_event_type                4 non-null      object
 37  marketing_event_target              0 non-null      float64
 38  customer_name                       425 non-null    object
 39  customer_type                       425 non-null    object
 40  customer_email                      425 non-null    object
 41  customer_id                         459 non-null    int64
 42  billing_postal_code                 360 non-null    object
 43  billing_country                     425 non-null    object
 44  billing_region                      0 non-null      float64
 45  billing_city                        425 non-null    object
 46  billing_company                     0 non-null      float64
 47  sale_line_type                      425 non-null    object
 48  sale_kind                           425 non-null    object
 49  purchase_option                     425 non-null    object
 50  order_name                          425 non-null    object
 51  order_id                            459 non-null    int64
 52  financial_status                    425 non-null    object 
 53  fulfillment_status                  425 non-null    object 
 54  cost_tracked                        425 non-null    object 
 55  cancelled                           425 non-null    object 
 56  gross_sales                         459 non-null    float64
 57  discounts                           459 non-null    float64
 58  returns                             459 non-null    float64
 59  shipping                            459 non-null    float64
 60  duties                              459 non-null    float64
 61  additional_fees                     459 non-null    float64
 62  taxes                               459 non-null    float64
 63  total_sales                         459 non-null    float64
 64  average_order_value                 459 non-null    float64
 65  percent_of_sales_with_staff_help    459 non-null    int64  
 66  returned_item_quantity              459 non-null    int64  
 67  average_units_ordered               459 non-null    int64  
 68  ordered_item_quantity               459 non-null    int64  
 69  net_quantity                        459 non-null    int64  
 70  pending_sales                       459 non-null    float64
 71  customers                           459 non-null    int64  
 72  units_per_transaction               459 non-null    int64  
 73  total_tips                          459 non-null    float64
 74  total_cost                          459 non-null    float64
 75  return_fees                         459 non-null    float64
 76  orders                              459 non-null    int64  
 77  net_sales                           459 non-null    float64
 78  gross_profit                        459 non-null    float64
 79  gross_margin                        459 non-null    float64
dtypes: float64(26), int64(14), object(40)
memory usage: 287.0+ KB
```

In [384]:

```
df3.info()
```

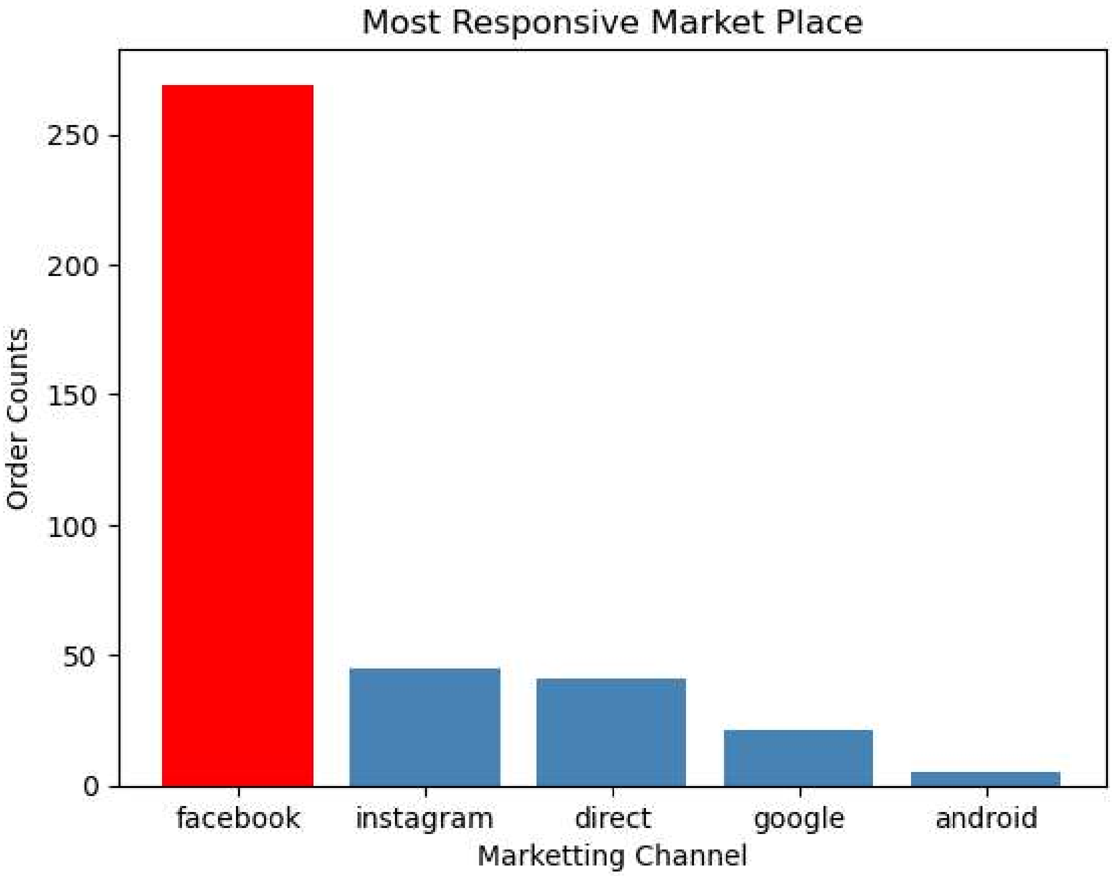
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 381 entries, 0 to 380
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   referring_channel      381 non-null    object 
 1   referring_category     338 non-null    object 
 2   fulfillment_status     381 non-null    object 
 3   order_id               381 non-null    int64  
 4   purchase_option        381 non-null    object 
 5   billing_city           381 non-null    object 
 6   customer_type          381 non-null    object 
 7   product_price          381 non-null    float64
 8   api_client_title       381 non-null    object 
 9   shipping_city          378 non-null    object 
 10  shipping_country       378 non-null    object 
 11  month                  381 non-null    object 
 12  day                    381 non-null    object 
 13  year                   381 non-null    int64  
 14  market_name            381 non-null    object 
 15  utm_campaign_content   305 non-null    object 
 16  utm_campaign_medium    286 non-null    object 
 17  orders                 381 non-null    int64  
 18  total_sales            381 non-null    float64
 19  net_sales              381 non-null    float64
 20  total_cost             381 non-null    float64
 21  shipping               381 non-null    float64
 22  units_per_transaction  381 non-null    int64  
dtypes: float64(5), int64(4), object(14)
memory usage: 68.6+ KB
```

In [385]:

```
referring_channel =df3["referring_channel"].value_counts()
```

In [386]:

```
colors = ['red', 'steelblue', 'steelblue','steelblue','steelblue','s
plt.bar(referring_channel.index, referring_channel.values,color=colo
plt.title('Most Responsive Market Place')
plt.xlabel('Marketting Channel')
plt.ylabel('Order Counts')
plt.show()
```



In [387]:

```
df3["shipping_city"].value_counts().head(10)
```

Out[387]:

```
shipping_city
Karachi                                            78
Islamabad                                          65
Lahore                                             51
Rawalpindi                                         21
Gilgit                                             20
Hyderabad                                          17
Peshawar                                           14
Faisalabad                                          8
Parachinar, District Kurram, Khyber Pakhtunkhwa     6
Swabi                                               6
Name: count, dtype: int64
```

In [388]:

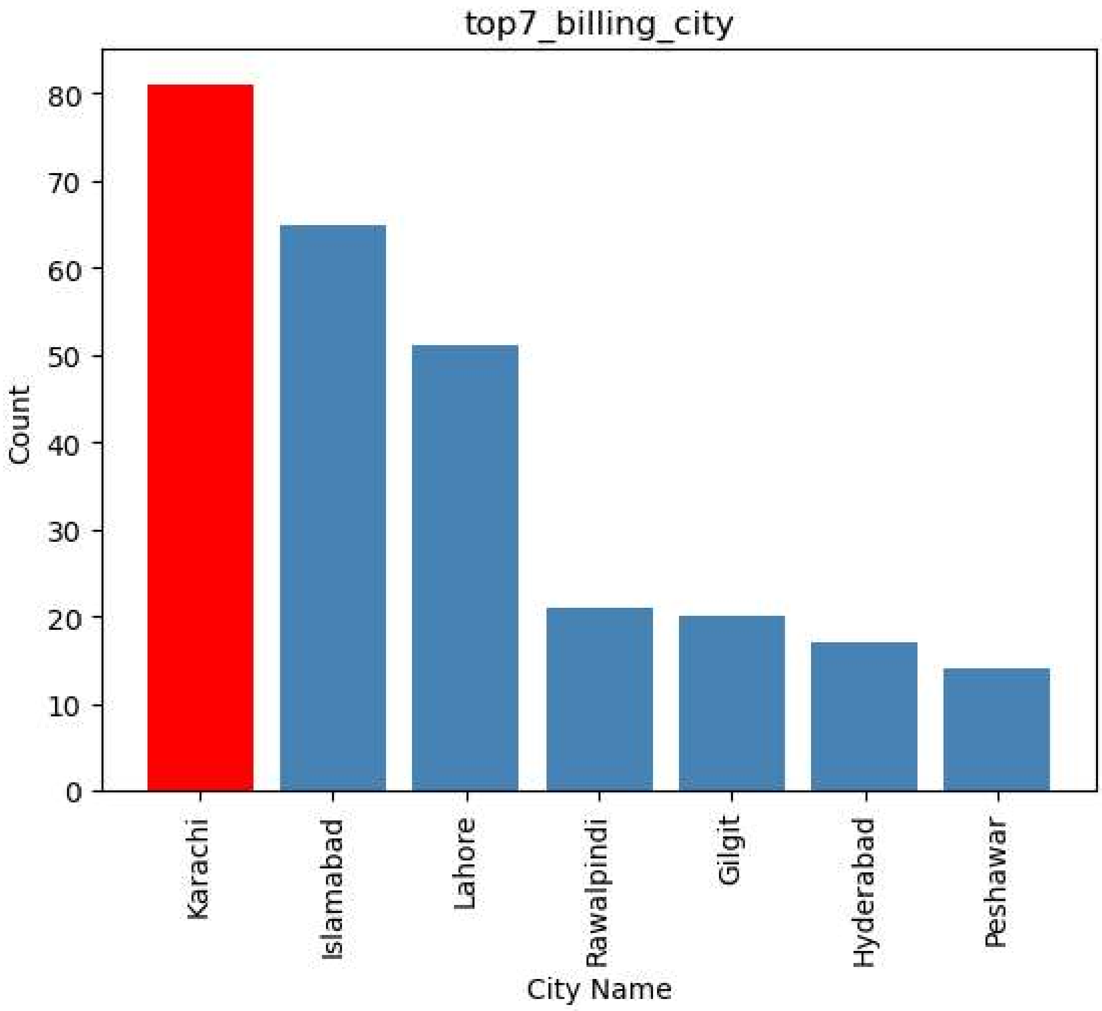
```
top7_billing_city = df3['billing_city'].value_counts().head(7)
top7_billing_city
```

Out[388]:

```
billing_city
Karachi       81
Islamabad     65
Lahore        51
Rawalpindi    21
Gilgit        20
Hyderabad     17
Peshawar      14
Name: count, dtype: int64
```

In [389]:

```
# Plotting
colors = ['red', 'steelblue', 'steelblue','steelblue','steelblue','s
plt.bar(top7_billing_city.index, top7_billing_city.values,color=colo
plt.xticks(rotation=90)
plt.title('top7_billing_city')
plt.xlabel('City Name')
plt.ylabel('Count')
plt.show()
```



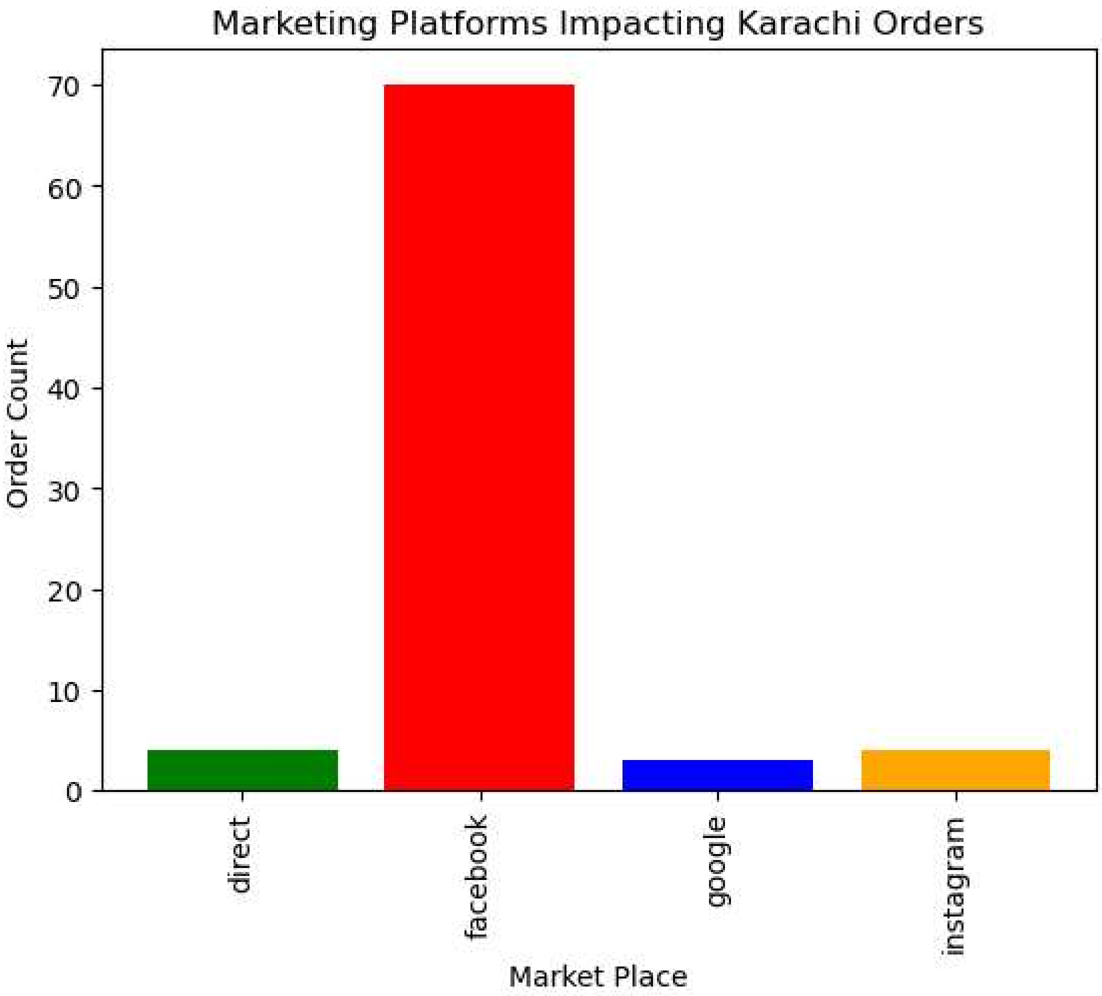
In [390]:

```
filtered_df = df3[df3["billing_city"] == "Karachi"]

# Grouping by referring channel
grouped_data = filtered_df.groupby(df3["referring_channel"]).size()

print(grouped_data)
```

```
referring_channel
direct        4
facebook     70
google        3
instagram     4
dtype: int64
```

In [391]:

```
colors = [ 'green','red', 'blue', 'orange','purple']
plt.bar(grouped_data.index, grouped_data.values,color=colors)
plt.xticks(rotation=90)
plt.title('Marketing Platforms Impacting Karachi Orders')
plt.xlabel('Market Place')
plt.ylabel('Order Count')
plt.show()
```



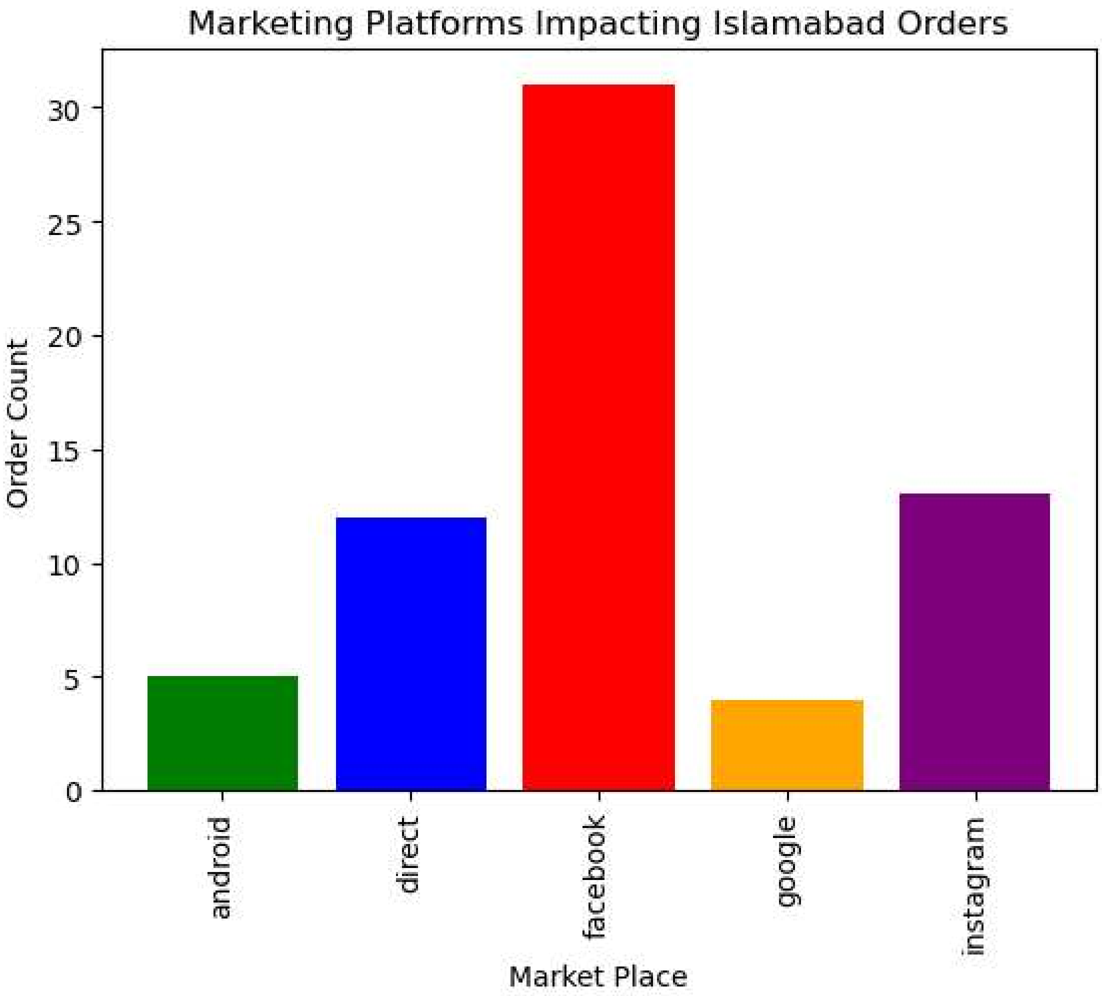
In [392]:

```
filtered_df = df3[df3["billing_city"] == "Islamabad"]

# Grouping by referring channel
grouped_data2 = filtered_df.groupby(df3["referring_channel"]).size()

print(grouped_data2)
```

```
referring_channel
android       5
direct       12
facebook     31
google        4
instagram    13
dtype: int64
```

In [393]:

```
colors = [ 'green', 'blue','red', 'orange','purple']
plt.bar(grouped_data2.index, grouped_data2.values, color=colors)
plt.xticks(rotation=90)
plt.title('Marketing Platforms Impacting Islamabad Orders')
plt.xlabel('Market Place')
plt.ylabel('Order Count')
plt.show()
```



Platform Traffic Analysis

In [394]:

```
df4.head()
```

Out[394]:

	day	adjustment	market_name	customer_cohort_quarter	customer_cohort_month	cust
0	2024-01-01	No	Pakistan	2024-01	2024-01	
1	2024-01-01	No	Pakistan	2024-01	2024-01	
2	2024-01-01	No	Pakistan	2024-01	2024-01	
3	2024-01-01	No	Pakistan	2024-01	2024-01	
4	2024-01-01	No	Pakistan	2024-01	2024-01	

5 rows × 80 columns

In [395]:

```
df4.drop(columns=["adjustment","market_name","customer_cohort_quarte

```

In [421]:

```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 188 entries, 0 to 457
Data columns (total 42 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   day                    188 non-null    object 
 1   referring_traffic      188 non-null    object 
 2   referring_category     175 non-null    object 
 3   referring_channel      188 non-null    object 
 4   referring_platform     188 non-null    object 
 5   referrer_source        188 non-null    object 
 6   referrer_name          144 non-null    object 
 7   shipping_city          188 non-null    object 
 8   api_client_title       188 non-null    object 
 9   product_price          188 non-null    float64
 10  utm_campaign_source    159 non-null    object 
 11  utm_campaign_medium    149 non-null    object 
 12  utm_campaign_content   156 non-null    object 
 13  customer_name          188 non-null    object 
 14  customer_type          188 non-null    object 
 15  customer_email         188 non-null    object 
 16  customer_id            188 non-null    int64  
 17  billing_country        188 non-null    object 
 18  billing_city           188 non-null    object 
 19  sale_line_type         188 non-null    object 
 20  sale_kind              188 non-null    object 
 21  purchase_option        188 non-null    object 
 22  order_name             188 non-null    object 
 23  order_id               188 non-null    int64  
 24  financial_status       188 non-null    object 
 25  fulfillment_status     188 non-null    object 
 26  cost_tracked           188 non-null    object 
 27  cancelled              188 non-null    object 
 28  gross_sales            188 non-null    float64
 29  discounts              188 non-null    float64
 30  returns                188 non-null    float64
 31  shipping               188 non-null    float64
 32  total_sales            188 non-null    float64
 33  net_quantity           188 non-null    int64  
 34  customers              188 non-null    int64  
 35  units_per_transaction  188 non-null    int64  
 36  total_cost             188 non-null    float64
 37  return_fees            188 non-null    float64
 38  orders                 188 non-null    int64  
 39  net_sales              188 non-null    float64
 40  gross_profit           188 non-null    float64
 41  gross_margin           188 non-null    float64
dtypes: float64(11), int64(6), object(25)
memory usage: 67.2+ KB
```

In [397]:

```
df4["sale_kind"].head(15)
```

Out[397]:

```
0      order
1      order
2     return
3      order
4      order
5      order
6      order
7      order
8      order
9     return
10     order
11    return
12     order
13     order
14     order
Name: sale_kind, dtype: object
```

In [398]:

```
df4 = df4[(df4['sale_kind'] != 'return')& df4['sale_kind'].notna()]
df4['sale_kind'].head(15)
```

Out[398]:

```
0     order
1     order
3     order
4     order
5     order
6     order
7     order
8     order
10    order
12    order
13    order
14    order
19    order
20    order
21    order
Name: sale_kind, dtype: object
```

In [399]:

```
df4=df4.drop(columns="referrer_url")

```

In [400]:

```
df4=df4.drop(columns="referrer_host")
```

In [401]:

```
df4=df4.drop(columns="variant_sku")
```

In [402]:

```
df4=df4.drop(columns="billing_postal_code")
```

In [403]:

```
df4 = df4[df4['sale_line_type'] == 'product']
```

In [404]:

```
df4['sale_line_type'].head()
```

Out[404]:

```
0    product
4    product
5    product
7    product
8    product
Name: sale_line_type, dtype: object
```

In [405]:

```
df4 = df4[df4['cancelled'] != 'Yes']
df4["cancelled"].head()
```

Out[405]:

```
0    No
4    No
5    No
7    No
8    No
Name: cancelled, dtype: object
```

In [406]:

```
df4["cancelled"].head(10)
```

Out[406]:

```
0     No
4     No
5     No
7     No
8     No
13    No
14    No
20    No
22    No
32    No
Name: cancelled, dtype: object
```

In [407]:

```
distinct_values = df4['customer_name'].unique()
distinct_values
```

Out[407]:

```
array(['Osama Ashfaq', 'Shazia Irfan', 'Babar Khan', 'Jamshed Talpur',
       'Farrukh Habib', 'Mrs shahzad Arain', 'Hassan Shahzad Anwar',
       'zulfiqar ali muhammad', 'Ghulam Sarwar', 'Asad Abbas Mirza',
       'Shaheryar Hussain', 'Omair Alavi', 'Muhammad Muhammad Basit',
       'Faisal Majeed', 'Rubina Feroz', 'Mehreen Malik', 'Amal Nabee
l',
       'Amber kamrankhan', 'Faheem Hashmi', 'muhammad Tariq',
       'Amjad Hayat', 'Mohsin Jamal', 'Mian Noorullah Noorullah',
       'Khurrum Jawaid', 'Mahwash Alavi', 'Zeeshan Zia', 'Shehzad Raso
ol',
       'Niaz Betab', 'Ali Akbar', 'Irum Zahra', 'Zaafir Khan',
       'Adeeb Rattar', 'Najam Khan', 'Muhammad Suhaib A. K. Bangash',
       'Fareed ud din Doctor', 'Haseeb Gul', 'Tahir Tasneem',
       'Abdullah Usmani', 'Dr Tehseen Iqbal Prof',
       'Dr Abbas Reza Hussain', 'Mrs Shahzad', 'Faiza Khan',
       'Zarmina Asad', 'Saifullah Sami', 'Tahirch Tasneem',
       'Tariq Afridi', 'Saira Fawad', 'Mahum Mohsin', 'Saad Kureshi',
       'asas asad', 'Dr. Iftikhar Qayum', 'Shaheer Mir', 'Raashid Janj
ua',
       'Khizra Arshid', 'Mushtaque Rasul Chaudhry Chaudhry', 'Farman K
tk',
       'Junaid Khan', 'muhammed umer lakhani', 'Dua Feroz',
       'Faryyal Kamran', 'Rauf Malik', 'Sehrish Malik', 'Irshad Ahme
d',
       'Mahesh Vaswani', 'Najam Sethi', 'Rafia Jamal', 'Azam Amir',
       'shoaib Qureshi', 'Waqas ahmad Pirjha', 'Muhammad Ali Hussain',
       'Mrs Utbah', 'Habib Ur Rehman', 'Waqar Ali Mahesar',
       'Haider Abbas', 'Dr Tanya Dogar', 'Sana Malik', 'Isma Ahmed',
       'Muhammad Ali', 'Ramsha Aamir', 'Ahmed khan Khan',
       'Manaksha Memon', 'Hina Siddiqui', 'Mahreen Pasha',
       'Haroon Rashid', 'Raja Hamza', 'Makhdoom Saifullah',
       'Aijaz Shaykh', 'poshmal ahmad', 'Dr Shahzad'], dtype=object)
```

In [408]:

```
grouped6 = df4.groupby('customer_name')['customer_type'].first().res
grouped6
```

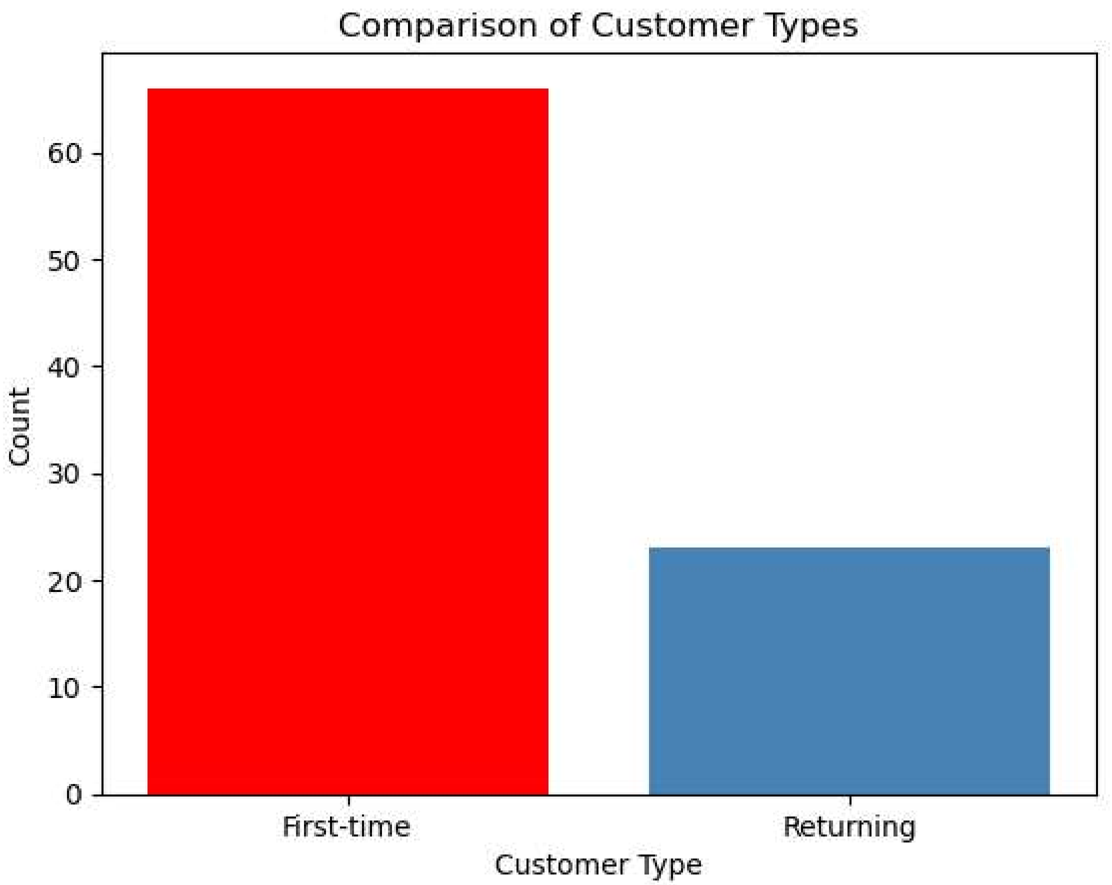
Out[408]:

	customer_name	customer_type
0	Abdullah Usmani	First-time
1	Adeeb Rattar	First-time
2	Ahmed khan Khan	First-time
3	Aijaz Shaykh	Returning
4	Ali Akbar	First-time
...	...	...
84	muhammad Tariq	Returning
85	muhammed umer lakhani	First-time
86	poshmal ahmad	First-time
87	shoaib Qureshi	First-time
88	zulfiqar ali muhammad	First-time

89 rows × 2 columns

In [422]:

```
colors = ['red', 'steelblue', 'steelblue','steelblue','steelblue','s
customer_counts = grouped6['customer_type'].value_counts()

# Plotting
plt.bar(customer_counts.index, customer_counts.values,color=colors)
plt.title('Comparison of Customer Types')
plt.xlabel('Customer Type')
plt.ylabel('Count')
#for i, value in enumerate(customer_counts):
    #plt.text(i, value /2, f'{(value/89)*100 :.2f}%', ha='center', v
plt.show()
```



In [410]:

```
df4["net_quantity"].value_counts()
```

Out[410]:

```
net_quantity
1    188
Name: count, dtype: int64
```

In [411]:

```
df4["referring_traffic"].value_counts()
```

Out[411]:

```
referring_traffic
paid       149
organic     17
direct      12
unknown     10
Name: count, dtype: int64
```

In [412]:

```
df4["referring_channel"].value_counts()
```

Out[412]:

```
referring_channel
facebook     137
instagram     24
direct        12
google        11
android        4
Name: count, dtype: int64
```

In [413]:

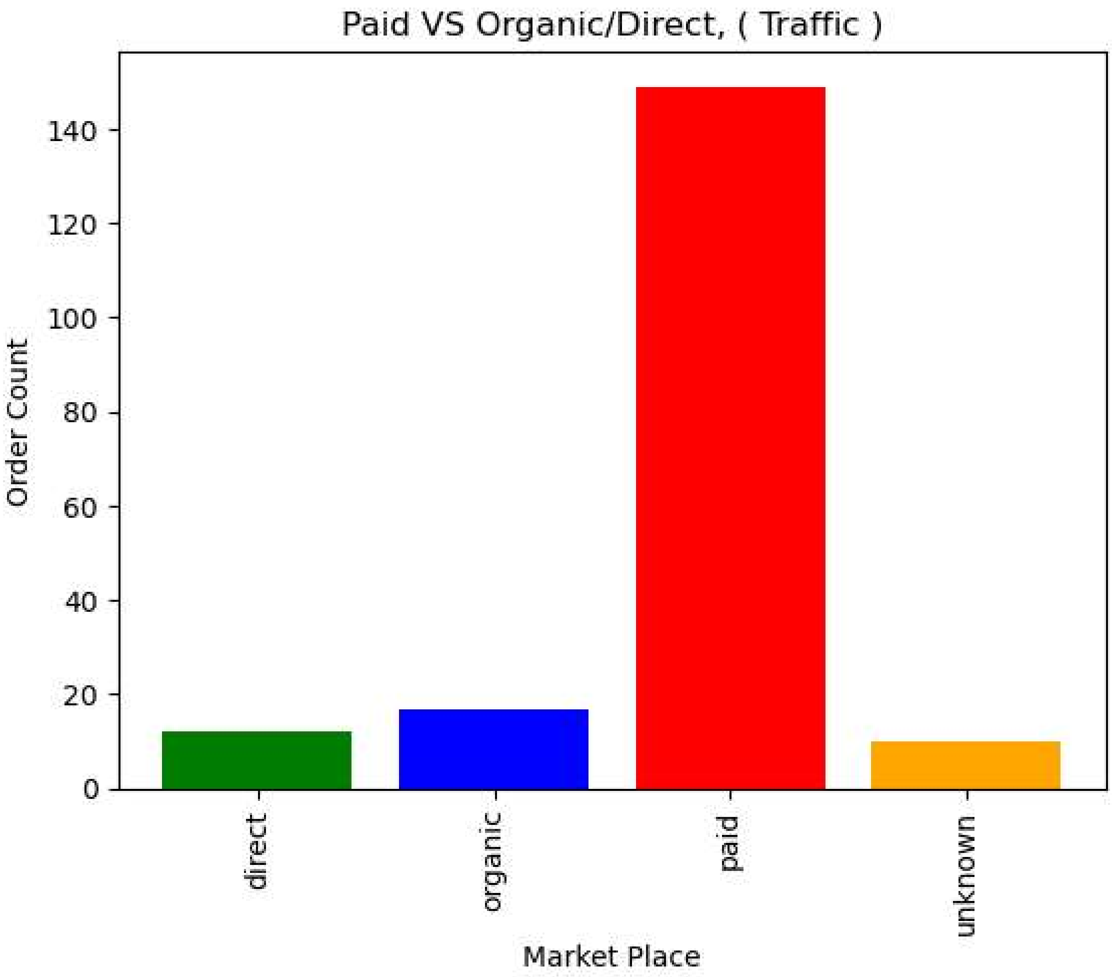
```
grouped_data5 = (df4["referring_channel"]).groupby(df4["referring_tr
grouped_data5
```

Out[413]:

```
referring_traffic
direct      12
organic     17
paid       149
unknown     10
Name: referring_channel, dtype: int64
```

In [414]:

```
colors = [ 'green', 'blue','red', 'orange','purple']
plt.bar(grouped_data5.index, grouped_data5.values, color=colors)
plt.xticks(rotation=90)
plt.title('Paid VS Organic/Direct, ( Traffic )')
plt.xlabel('Market Place')
plt.ylabel('Order Count')
plt.show()
```



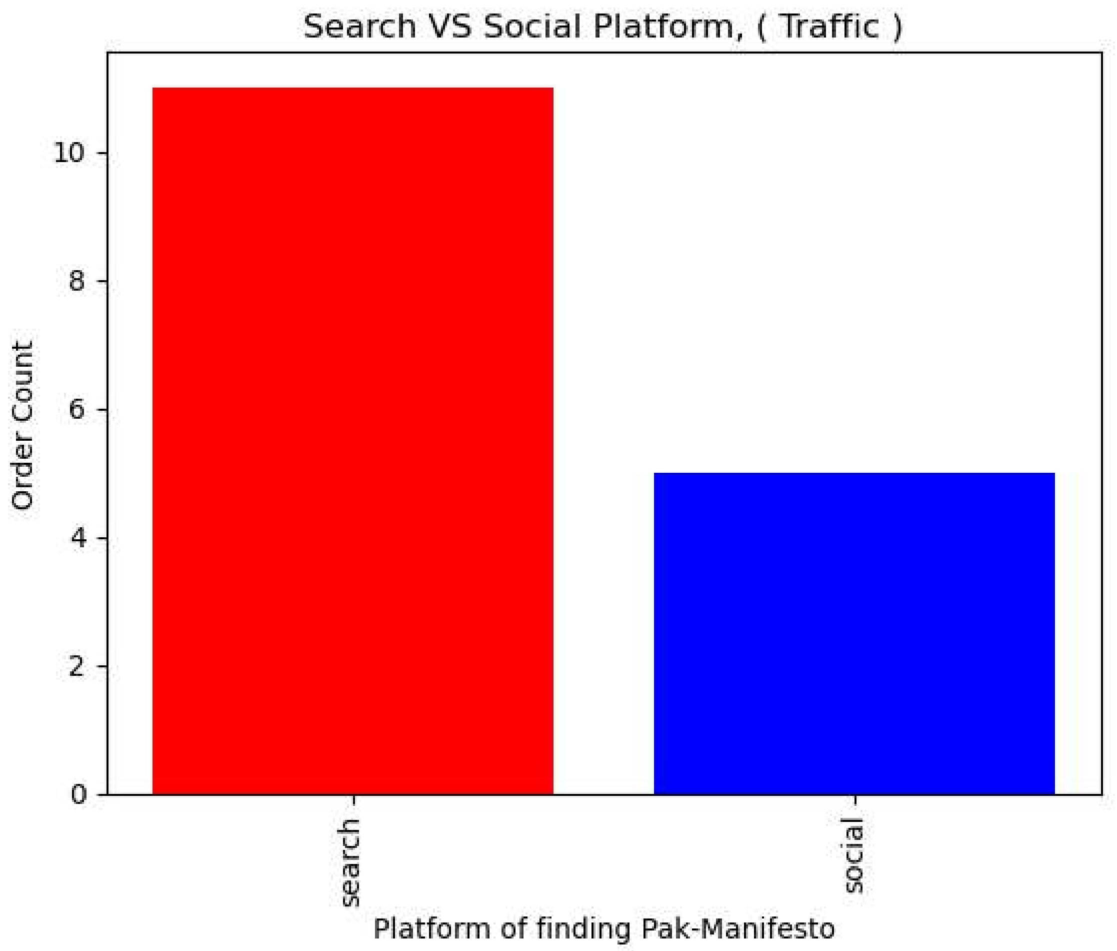
In [415]:

```
filtered_df = df4[df4["referring_traffic"] == "organic" ]

# Grouping by referring channel
grouped_data6 = filtered_df.groupby(df4["referring_category"]).size(

print(grouped_data6)
```

```
referring_category
search    11
social     5
dtype: int64
```

In [416]:

```
colors = [ 'red', 'blue','red', 'orange','purple']
plt.bar(grouped_data6.index, grouped_data6.values, color=colors)
plt.xticks(rotation=90)
plt.title('Search VS Social Platform, ( Traffic )')
plt.xlabel('Platform of finding Pak-Manifesto')
plt.ylabel('Order Count')
plt.show()
```



In [417]:

```
sale_kind= df4["sale_kind"].value_counts()
sale_kind
```

Out[417]:

```
sale_kind
order    188
Name: count, dtype: int64
```

In [ ]:

```

```

Marketing Insights Report for PAK-MANIFESTO ENTERPRISES

Key Findings:

Effective Facebook Marketing:

The primary source of orders for PAK-MANIFESTO stems from its Facebook paid marketing campaigns, underscoring the efficacy of this channel in driving sales and customer engagement.

Underperformance of Instagram Marketing:

In contrast to the success of Facebook marketing efforts, Instagram paid marketing has not yielded significant results in generating orders for the business, highlighting an area for potential optimization or reevaluation of strategy.

Opportunity for SEO Enhancement:

The limited impact of organic orders suggests a need for enhancement in PAK-MANIFEST's Search Engine Optimization (SEO) strategy. This presents an opportunity to bolster the visibility and reach of the company's offerings in online search results.

Strong Customer Retention:

Analysis indicates a commendable rate of returning customers, reflecting positively on the brand's ability to maintain customer loyalty and satisfaction over time.

Optimal Cities for Facebook Campaigns:

Karachi, Islamabad, and Lahore emerge as the top-performing cities for Facebook campaigns, indicating the effectiveness of targeted marketing efforts in these metropolitan areas. Recommendations:

Maximize Facebook Marketing Impact:

Continue to invest resources and efforts in refining and expanding Facebook marketing strategies to sustain and potentially amplify the current success rate.

Reevaluate Instagram Marketing Approach:

Assess the underlying reasons for the underperformance of Instagram marketing campaigns and explore alternative tactics or adjustments to improve their effectiveness.

Enhance SEO Strategy:

Implement measures to strengthen the SEO strategy, including keyword optimization, content enhancements, and backlink building, to increase organic visibility and drive higher traffic volumes.

Nurture Customer Relationships:

PAK-MANIFEST boasts a commendable customer return rate, indicating a strong level of customer satisfaction and loyalty. This positive metric underscores the business's ability to consistently meet or exceed customer expectations, fostering a sense of trust and reliability among its clientele. With a high rate of returning customers, PAK-MANIFEST not only demonstrates the quality of its products or services but also reflects its success in nurturing long-term relationships with its audience. This repeat business is not only a testament to the exceptional value proposition offered by PAK-MANIFEST but also serves as a solid foundation for sustained growth and success in the competitive marketplace

Focus on Targeted Geographic Campaigns:

Direct marketing efforts towards the most responsive cities, such as Karachi, Islamabad, and Lahore, while also exploring opportunities to replicate successful strategies in other key markets.

Conclusion

By aligning marketing efforts with these insights and recommendations, PAK-MANIFESTO can further optimize its marketing strategies, enhance customer engagement, and drive sustainable business growth.

In [418]:

```
# Drop rows based on conditions
# For example, to drop rows where a column 'age' has values less tha
```