Problem statment

To help fintech startups better understand customer behavior and optimize their UPI-based digital payment services, this project analyzes a dataset of 250,000 UPI transactions from 2024. The goal is to uncover trends in user activity, device preferences, age demographics, transaction timing, and merchant category usage. These insights will enable product, marketing, and operations teams to make data-driven decisions that enhance user experience, drive engagement, and increase UPI adoption.

In [28]:
```
# Importing ibraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [29]:
```
#Reading data
df=pd.read_csv('upi_transactions_2024.csv')
```

Data Overview

In [30]:
```
df.head(3)
```

Out[30]:

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_sta
0	TXN0000000001	2024-10-08 15:17:28	P2P	Entertainment	868	SUCC
1	TXN0000000002	2024-04-11 06:56:00	P2M	Grocery	1011	SUCC
2	TXN0000000003	2024-04-02 13:27:18	P2P	Grocery	477	SUCC

In [31]:
```
df.shape
```

Out[31]:
```
(250000, 17)
```

In [32]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250000 entries, 0 to 249999
Data columns (total 17 columns):
 #   Column              Non-Null Count   Dtype 
---  ------              --------------   ----- 
 0   transaction id      250000 non-null  object
 1   timestamp           250000 non-null  object
 2   transaction type    250000 non-null  object
 3   merchant_category   250000 non-null  object
 4   amount (INR)        250000 non-null  int64 
 5   transaction_status  250000 non-null  object
 6   sender_age_group    250000 non-null  object
 7   receiver_age_group  250000 non-null  object
 8   sender_state        250000 non-null  object
 9   sender_bank         250000 non-null  object
 10  receiver_bank       250000 non-null  object
 11  device_type         250000 non-null  object
 12  network_type        250000 non-null  object
 13  fraud_flag          250000 non-null  int64 
 14  hour_of_day         250000 non-null  int64 
 15  day_of_week         250000 non-null  object
 16  is_weekend          250000 non-null  int64 
dtypes: int64(4), object(13)
memory usage: 32.4+ MB
```

In [33]:
```
#changing the data type
df['timestamp']=pd.to_datetime(df['timestamp'])
```

In [34]:
```
df[df.duplicated()]
```

Out[34]:

transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status

In [35]:
```
df.isnull().sum()
```

Out[35]:

	0
transaction id	0
timestamp	0
transaction type	0
merchant_category	0
amount (INR)	0
transaction_status	0
sender_age_group	0
receiver_age_group	0
sender_state	0
sender_bank	0
receiver_bank	0
device_type	0
network_type	0
fraud_flag	0
hour_of_day	0
day_of_week	0
is_weekend	0

dtype: int64

In [36]:

```
list=['transaction type', 'merchant_category','transaction_status','hour_of_day'
       'sender_state', 'sender_bank', 'receiver_bank', 'device_type',
       'network_type','is_weekend','day_of_week']
for i in list:
  print(i)
  print(f'Value_counts',df[i].value_counts())
  print('-'*50)
```

```
transaction type
Value_counts transaction type
P2P            112445
P2M             87660
Bill Payment    37368
Recharge        12527
Name: count, dtype: int64
--------------------------------------------------
merchant_category
Value_counts merchant_category
Grocery          49966
Food             37464
Shopping         29872
Fuel             25063
Other            24828
Utilities        22338
Transport        20105
Entertainment    20103
Healthcare       12663
Education         7598
Name: count, dtype: int64
--------------------------------------------------
transaction_status
Value_counts transaction_status
SUCCESS    237624
FAILED      12376
Name: count, dtype: int64
--------------------------------------------------
hour_of_day
Value_counts hour_of_day
19    21232
18    20064
20    18506
17    18340
12    17516
11    16328
21    16253
13    15038
16    13992
10    13904
15    12624
14    11472
9     10450
22     9364
8      8349
23     5817
7      5630
6      3501
0      3388
1      2244
5      1742
2      1685
3      1314
4      1247
Name: count, dtype: int64
--------------------------------------------------
sender_age_group
Value_counts sender_age_group
26-35    87432
36-45    62873
```

```
18-25    62345
46-55    24841
56+      12509
Name: count, dtype: int64
--------------------------------------------------
receiver_age_group
Value_counts receiver_age_group
26-35    87864
18-25    62611
36-45    62151
46-55    24823
56+      12551
Name: count, dtype: int64
--------------------------------------------------
sender_state
Value_counts sender_state
Maharashtra       37427
Uttar Pradesh     30125
Karnataka         29756
Tamil Nadu        25367
Delhi             24870
Telangana         22435
Gujarat           20061
Andhra Pradesh    20006
Rajasthan         19981
West Bengal       19972
Name: count, dtype: int64
--------------------------------------------------
sender_bank
Value_counts sender_bank
SBI         62693
HDFC        37485
ICICI       29769
IndusInd    25173
Axis        25042
PNB         24946
Yes Bank    24860
Kotak       20032
Name: count, dtype: int64
--------------------------------------------------
receiver_bank
Value_counts receiver_bank
SBI         62378
HDFC        37651
ICICI       29944
IndusInd    25086
Yes Bank    25009
Axis        24992
PNB         24802
Kotak       20138
Name: count, dtype: int64
--------------------------------------------------
device_type
Value_counts device_type
Android    187777
iOS         49613
Web         12610
Name: count, dtype: int64
--------------------------------------------------
network_type
```

```
Value_counts network_type
4G      149813
5G       62582
WiFi     25134
3G       12471
Name: count, dtype: int64
--------------------------------------------------
is_weekend
Value_counts is_weekend
0    178663
1     71337
Name: count, dtype: int64
--------------------------------------------------
day_of_week
Value_counts day_of_week
Monday       36495
Sunday       36003
Wednesday    35700
Tuesday      35540
Friday       35496
Thursday     35432
Saturday     35334
Name: count, dtype: int64
--------------------------------------------------
```

In [37]:

```
for i in list:
  print(i)
  print(f'Value_counts',df[i].unique())
  print('-'*50)
```

```
transaction type
Value_counts ['P2P' 'P2M' 'Bill Payment' 'Recharge']
--------------------------------------------------
merchant_category
Value_counts ['Entertainment' 'Grocery' 'Fuel' 'Shopping' 'Food' 'Other' 'Utiliti
es'
 'Transport' 'Healthcare' 'Education']
--------------------------------------------------
transaction_status
Value_counts ['SUCCESS' 'FAILED']
--------------------------------------------------
hour_of_day
Value_counts [15  6 13 10 19 22 18  9 20  0 12  7 17  4 21 16  1 14  5  8 11 23
3  2]
--------------------------------------------------
sender_age_group
Value_counts ['26-35' '36-45' '46-55' '56+' '18-25']
--------------------------------------------------
receiver_age_group
Value_counts ['18-25' '26-35' '36-45' '46-55' '56+']
--------------------------------------------------
sender_state
Value_counts ['Delhi' 'Uttar Pradesh' 'Karnataka' 'Telangana' 'Maharashtra' 'Guja
rat'
 'Rajasthan' 'Tamil Nadu' 'West Bengal' 'Andhra Pradesh']
--------------------------------------------------
sender_bank
Value_counts ['Axis' 'ICICI' 'Yes Bank' 'IndusInd' 'HDFC' 'Kotak' 'SBI' 'PNB']
--------------------------------------------------
receiver_bank
Value_counts ['SBI' 'Axis' 'PNB' 'Yes Bank' 'IndusInd' 'HDFC' 'Kotak' 'ICICI']
--------------------------------------------------
device_type
Value_counts ['Android' 'iOS' 'Web']
--------------------------------------------------
network_type
Value_counts ['4G' '5G' 'WiFi' '3G']
--------------------------------------------------
is_weekend
Value_counts [0 1]
--------------------------------------------------
day_of_week
Value_counts ['Tuesday' 'Thursday' 'Sunday' 'Monday' 'Saturday' 'Wednesday' 'Frid
ay']
--------------------------------------------------
```

Feature engineering

In [38]:
```python
# I dont want to touch the original data so we create a copy of it
dfn=df
```

In [39]:
```python
# hour_of_day can be set to Morning to Night
def get_time_of_day(hours):
  if 5 <= hours < 12:
    return 'Morning'
  elif 12<= hours < 17:
    return 'Afternoon'
  elif 17<=hours< 21:
```

```
        return 'Evening'
    else:
        return 'Night'

dfn['hour_of_day']=dfn['hour_of_day'].apply(get_time_of_day)
```

In [40]:
```
# In device_type we can make Android ,iOS as Phone and web as desktop
def device(type):
    if type in ['Android','iOS']:
        return 'Phone'
    else:
        return 'desktop'


dfn['device_type']=dfn['device_type'].apply(device)
```

In [41]:
```
# in is_weekend column 0= Weekday and 1= weekend
def days(num):
    if num==0:
        return 'Weekday'
    else:
        return 'Weekend'

dfn['is_weekend']=df['is_weekend'].apply(days)
```

Q1 When do users transact the most?

In [42]:
```
dfn.groupby('hour_of_day')['transaction id'].count()
```

Out[42]:

hour_of_day	transaction id
Afternoon	70642
Evening	78142
Morning	59904
Night	41312

dtype: int64

In [43]:
```
dfn.groupby('day_of_week')['transaction id'].count()
```

Out[43]:

	transaction id
day_of_week	
Friday	35496
Monday	36495
Saturday	35334
Sunday	36003
Thursday	35432
Tuesday	35540
Wednesday	35700

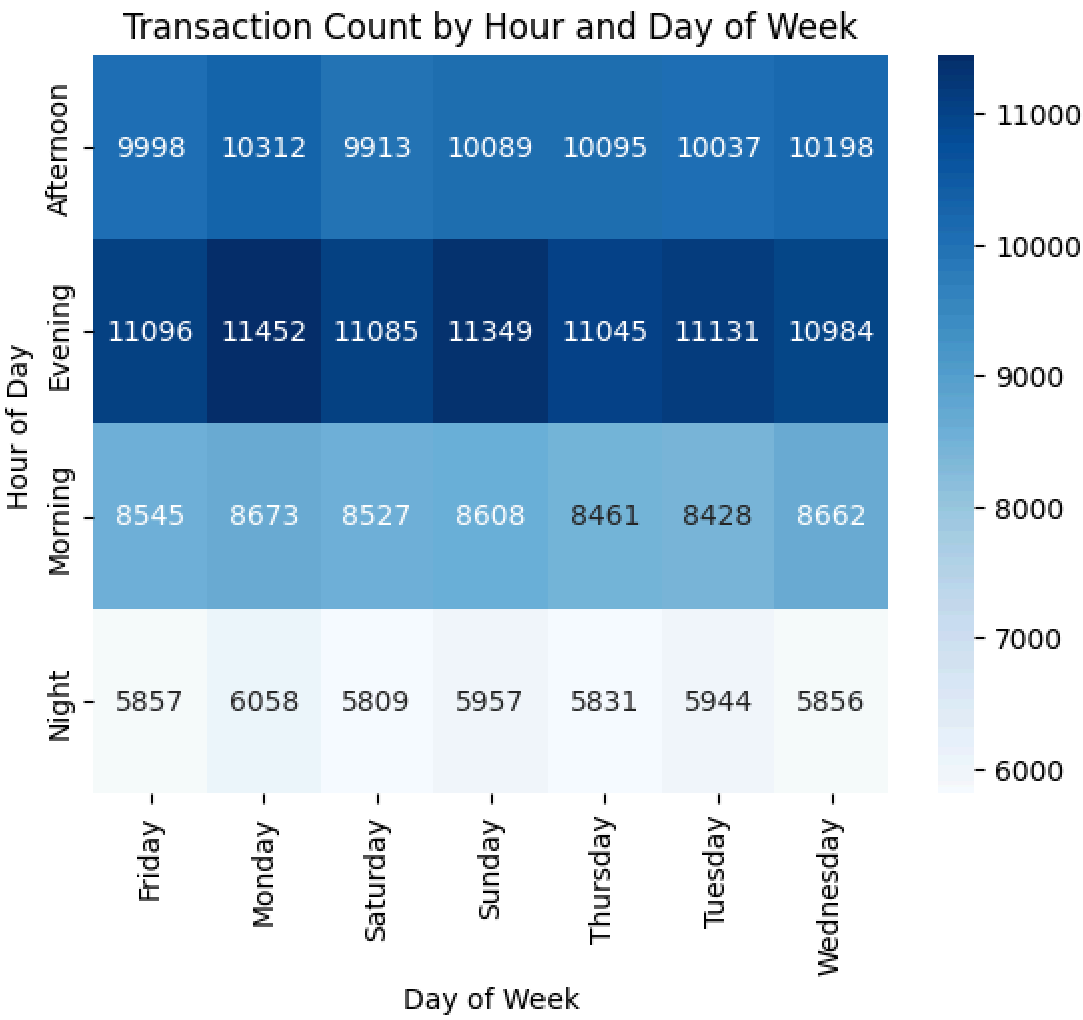
dtype: int64

In [44]:
```
pd.crosstab(dfn['hour_of_day'], dfn['day_of_week'])
```

Out[44]:

day_of_week	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
hour_of_day							
Afternoon	9998	10312	9913	10089	10095	10037	10198
Evening	11096	11452	11085	11349	11045	11131	10984
Morning	8545	8673	8527	8608	8461	8428	8662
Night	5857	6058	5809	5957	5831	5944	5856

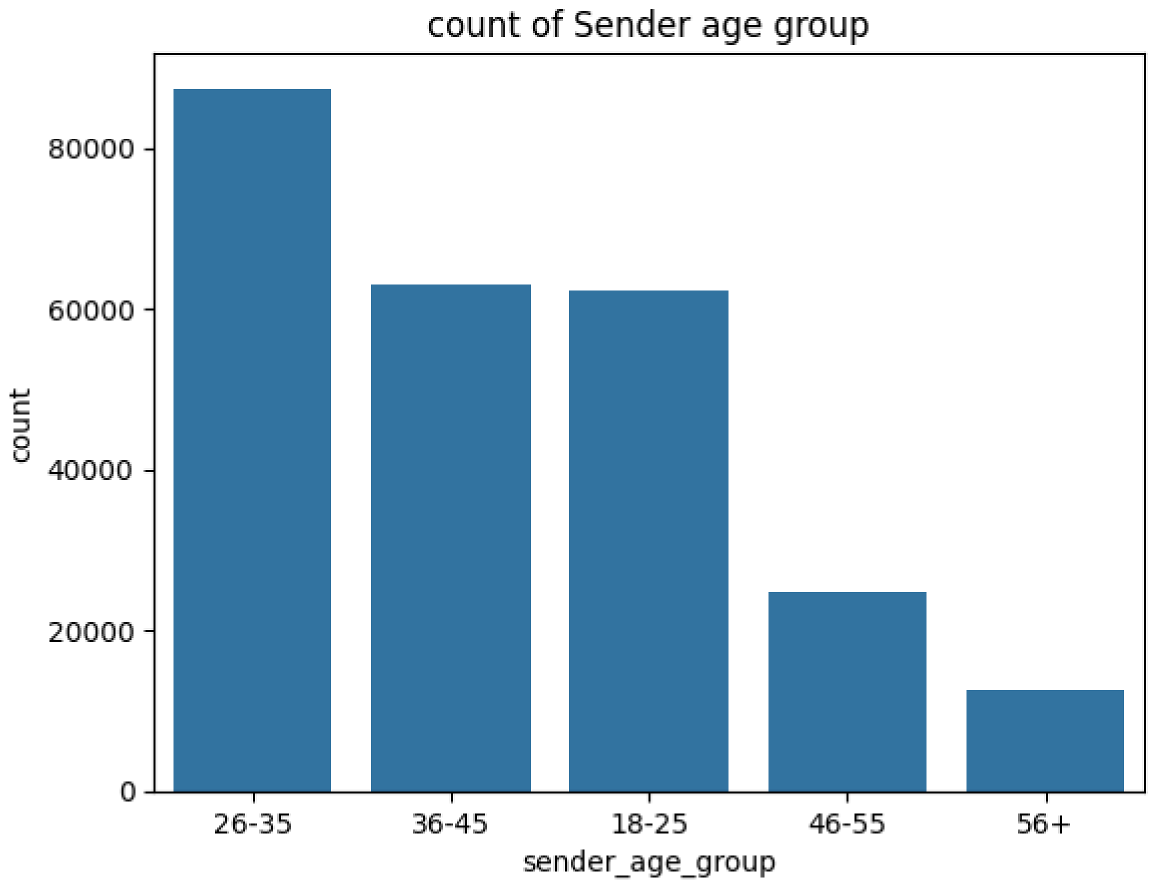
In [45]:
```
# we can use heatmap to see the exact hour and day of maximum use of UPI
heatmap_data = pd.crosstab(dfn['hour_of_day'], dfn['day_of_week'])
sns.heatmap(heatmap_data, cmap="Blues", annot=True, fmt='d')
plt.title("Transaction Count by Hour and Day of Week")
plt.xlabel("Day of Week")
plt.ylabel("Hour of Day")
plt.show()
```



- The highest number of transactions happen on Monday evenings.
- The lowest number of transactions occur on Saturday nights.
- Across all days, evenings consistently have the most activity, while nights see the least.

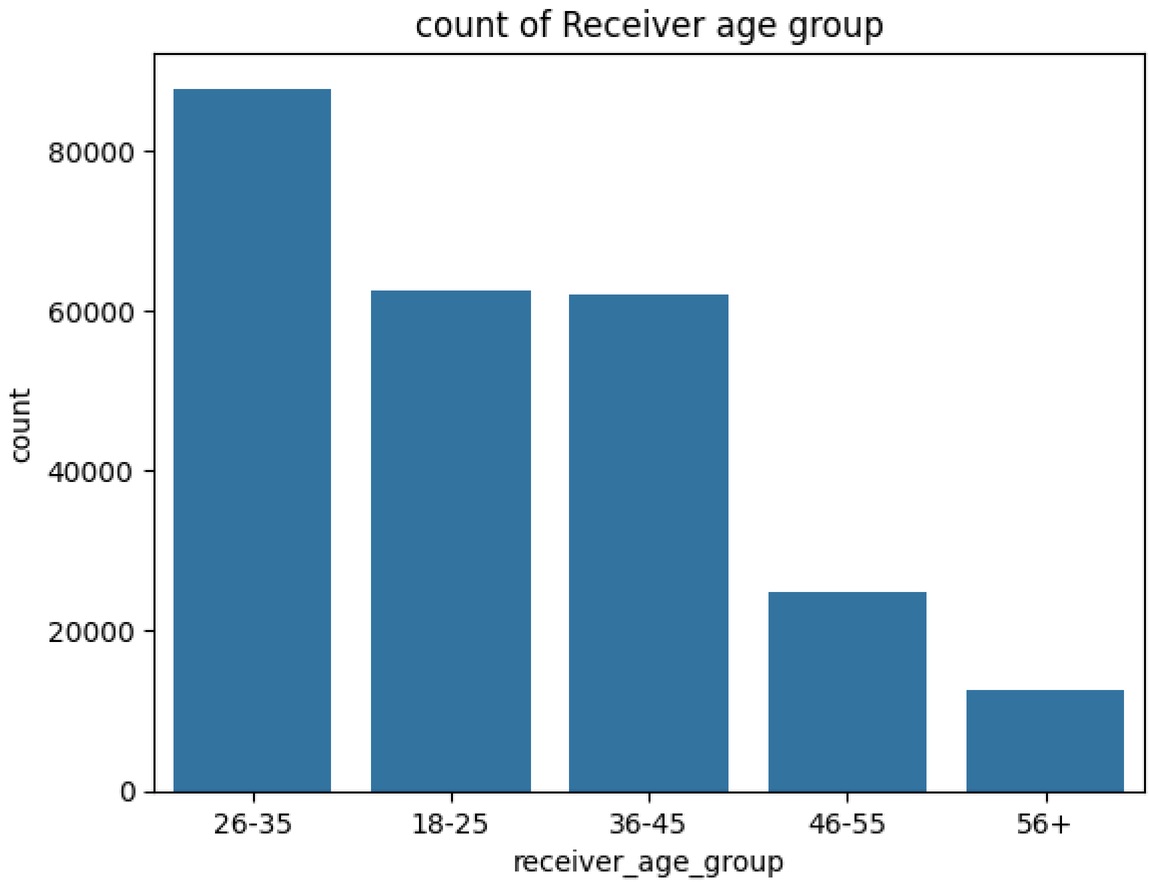
Q2 Which age groups are most active?

In [51]:
```
# Seeing sender_age_group
count_sender_age_group=dfn['sender_age_group'].value_counts()
count_sender_age_group=pd.DataFrame(count_sender_age_group)
sns.barplot(data=count_sender_age_group,x='sender_age_group',y='count')
plt.title('count of Sender age group')
plt.xlabel('sender_age_group')
plt.show()
```



In [52]:

```
# seeing receiver_age_group
count_receiver_age_group=dfn['receiver_age_group'].value_counts()
count_receiver_age_group=pd.DataFrame(count_receiver_age_group)
sns.barplot(data=count_receiver_age_group,x='receiver_age_group',y='count')
plt.title('count of Receiver age group')
plt.xlabel('receiver_age_group')
plt.show()
```



- The 26–35 age group is the most active in both sending and receiving UPI payments — likely due to higher spending habits and financial independence.
- The 56+ age group is the least active, possibly due to lower digital adoption or spending needs.

Q3 Which devices and network types are preferred?

In [53]:
```
dfn.groupby('device_type')['transaction id'].count()
```

Out[53]:

device_type	transaction id
Phone	237390
desktop	12610

dtype: int64

In [54]:
```
# Let us check with network type also

dfn.groupby('device_type')['network_type'].value_counts()
```

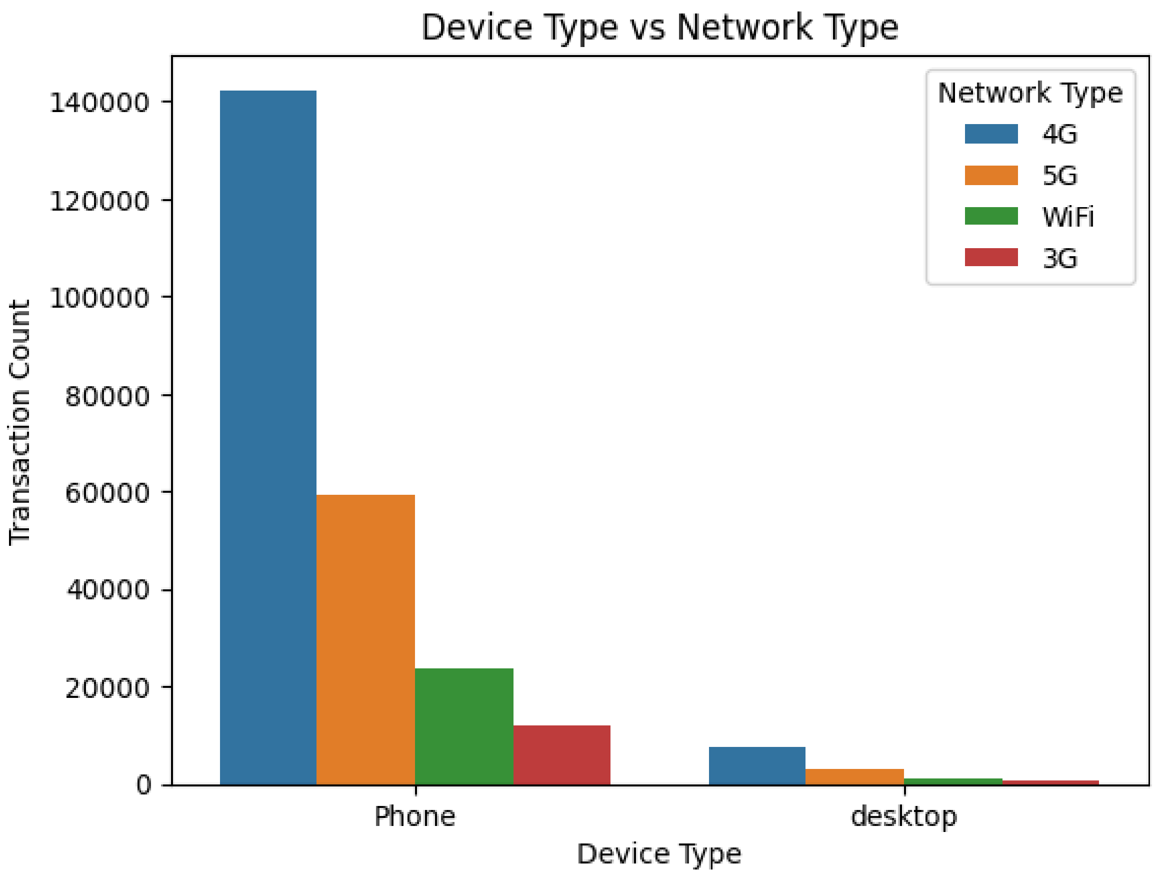
Out[54]:

device_type	network_type	count
Phone	4G	142294
	5G	59417
	WiFi	23844
	3G	11835
desktop	4G	7519
	5G	3165
	WiFi	1290
	3G	636

dtype: int64

In [55]:

```
sns.countplot(data=dfn, x='device_type', hue='network_type')
plt.title("Device Type vs Network Type")
plt.xlabel("Device Type")
plt.ylabel("Transaction Count")
plt.legend(title='Network Type')
plt.show()
```



- When grouped, phones (Android + iOS) are vastly more preferred than desktops for UPI transactions.

- Most users transact over 4G networks, suggesting high mobile usage and penetration.
- Desktop usage may be limited to net banking or card payments, rather than UPI.

Q4 Which merchant categories are the most popular?

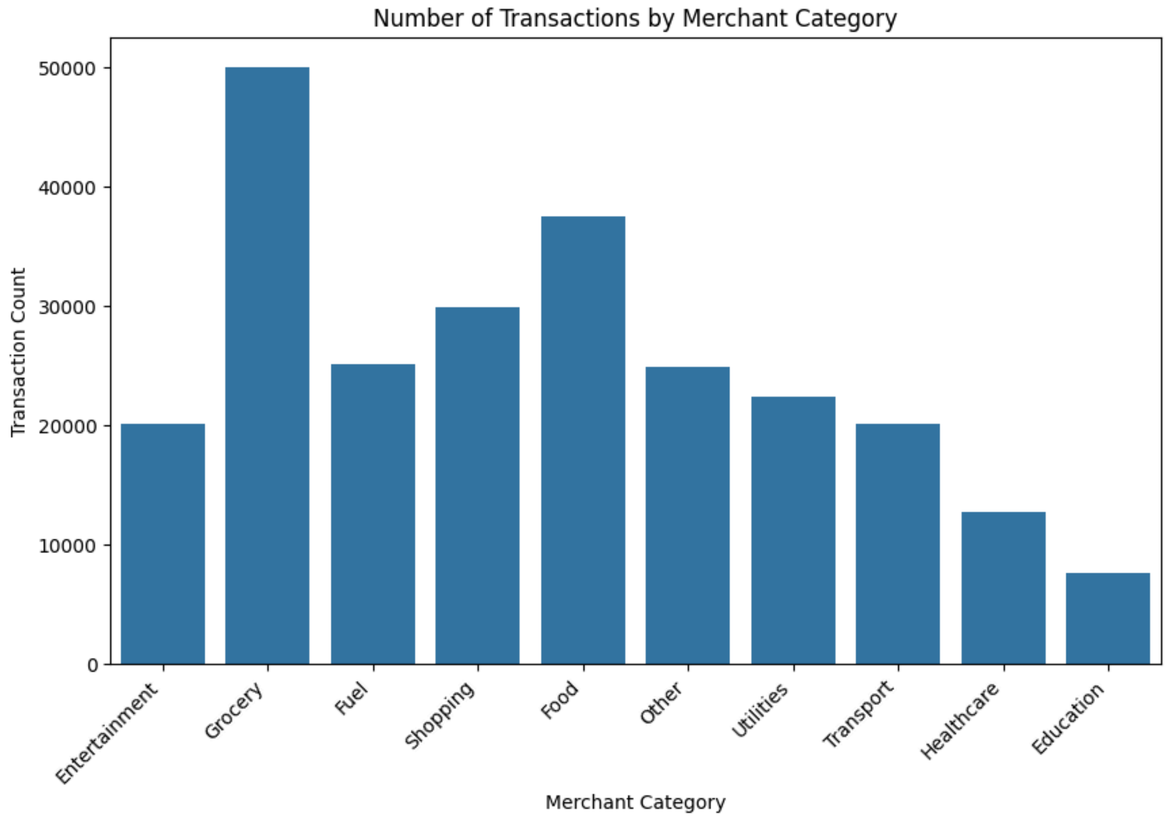
In [56]:
```
dfn.groupby('merchant_category')['transaction id'].count()
```

Out[56]:

merchant_category	transaction id
Education	7598
Entertainment	20103
Food	37464
Fuel	25063
Grocery	49966
Healthcare	12663
Other	24828
Shopping	29872
Transport	20105
Utilities	22338

dtype: int64

In [57]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=dfn, x='merchant_category')
plt.title("Number of Transactions by Merchant Category")
plt.xlabel("Merchant Category")
plt.ylabel("Transaction Count")
plt.xticks(rotation=45, ha='right')
plt.show()
```

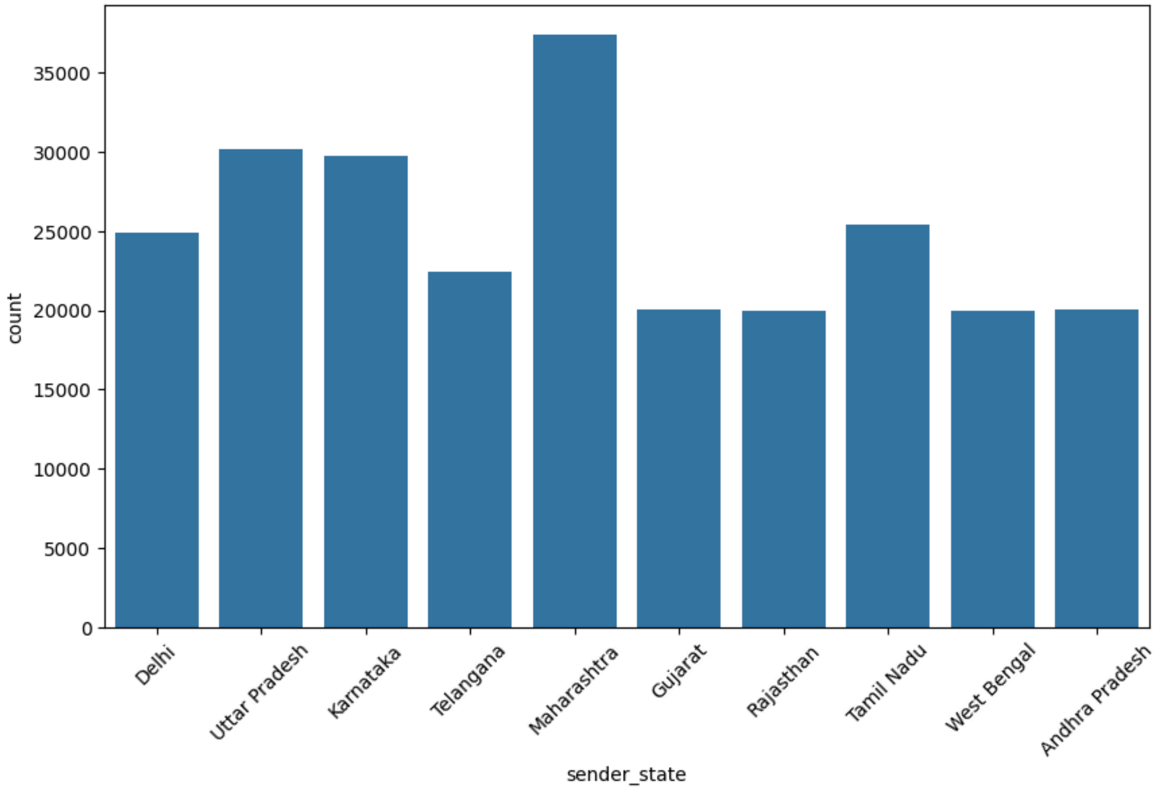


- Grocery is the most common merchant category for UPI transactions — likely due to frequent, low-value purchases.
- Education is the least common — possibly because high-value payments are not feasible via UPI due to limits or EMI-based structures.

Q5 Which states or banks are the most active?

In [58]:

```
plt.figure(figsize=(10,6))
sns.countplot(data=dfn,x='sender_state')
plt.xticks(rotation=45)
plt.show()
```



In [59]:
```
# receiver_bank
dfn['receiver_bank'].value_counts()
```

Out[59]:

	count
receiver_bank	
SBI	62378
HDFC	37651
ICICI	29944
IndusInd	25086
Yes Bank	25009
Axis	24992
PNB	24802
Kotak	20138

dtype: int64

In [60]:
```
# sender_bank
dfn['sender_bank'].value_counts()
```

Out[60]:

sender_bank	count
SBI	62693
HDFC	37485
ICICI	29769
IndusInd	25173
Axis	25042
PNB	24946
Yes Bank	24860
Kotak	20032

dtype: int64

Insights

- The highest number of transactions happen on **Monday evenings** and The lowest on **Saturday nights**.
- Across all days, **evenings consistently show the most activity**, while **nights see the least**.
- The **26–35 age group** is the most active in both sending and receiving UPI payments — likely due to **higher spending habits** and **financial independence**.
- The **56+ age group** is the least active, possibly due to **lower digital adoption** or **reduced financial activity**.
- **Mobile devices (Android + iOS)** are overwhelmingly preferred for UPI transactions over desktop (Web).
- Most transactions occur over **4G networks**, highlighting strong **mobile-first behavior**.
- **Grocery** is the most common category, likely due to **frequent low-value transactions**.
- **Education** sees the least usage, possibly due to **high costs** or **UPI transaction limits**.

Recomendations

- Run time-specific campaigns Use low-activity hours (like Saturday night) for backend updates or testing.
- Introduce features tailored to 26–35 (e.g., personal finance tracking, instant credit, split bills). Provide tutorials or simplified modes for 56+ users to improve adoption.
- Prioritize mobile-first designs and in-app UPI features. Since Android and iOS dominate, investing in lightweight and intuitive apps can directly increase transactions.

- Ensure the app performs well on mid-range devices and varying network speeds — especially in rural or Tier 2/3 cities.
- Encourage adoption in low-UPI sectors like Education by raising limits or supporting EMI-like features. Partner with educational institutions or healthcare providers.