In [ ]:

```
! pip install seaborn
! pip install numpy
! pip install pandas
! pip install matplotlib
```

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

In [2]:

```
# import csv file
df = pd.read_csv(r'C:\Users\Admin\Downloads\Projects-20260207T140253Z-1-001\Proj
```

In [3]:

```
#Check Number of rows and Columns
df.shape
```

Out[3]:

```
(11257, 13)
```

In [4]:

```
#To see the imported data
df.head(10)
```

Out[4]:

| | User_ID | Cust_name | Product_ID | Age | Age Group | Gender | State | Zone |
|---|---|---|---|---|---|---|---|---|
| **0** | 1002903.0 | Anvi | P00125942 | 27.0 | 26-35 | Female | Maharashtra | West |
| **1** | 1000732.0 | Shanta | P00110942 | 34.0 | 26-35 | Female | Andhra Pradesh | South |
| **2** | 1001990.0 | Sheetal | P00118542 | 16.0 | 0-17 | Female | Uttar Pradesh | Central |
| **3** | 1001425.0 | Virendra | P00237842 | 16.0 | 0-17 | M | Karnataka | South |
| **4** | 1000588.0 | Vishal | P00057942 | 28.0 | 26-35 | M | Gujarat | West |
| **5** | 1000588.0 | Suuraj | P00057942 | 28.0 | 26-35 | M | Himachal Pradesh | Northern |
| **6** | 1001132.0 | Sejal | P00018042 | 25.0 | 18-25 | Female | Uttar Pradesh | Central |
| **7** | 1002092.0 | Shivangi | P00273442 | 61.0 | 55+ | Female | Maharashtra | West |
| **8** | 1003224.0 | Kushal | P00205642 | 35.0 | 26-35 | M | Uttar Pradesh | Central |
| **9** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

In [5]:

```
#Field details and Data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11257 entries, 0 to 11256
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11245 non-null  float64
 1   Cust_name         11245 non-null  object 
 2   Product_ID        11245 non-null  object 
 3   Age               11245 non-null  float64
 4   Age Group         11245 non-null  object 
 5   Gender            11245 non-null  object 
 6   State             11245 non-null  object 
 7   Zone              11245 non-null  object 
 8   Zipcode           0 non-null      float64
 9   Profession        11245 non-null  object 
 10  Product_Category  11245 non-null  object 
 11  Orders            11245 non-null  float64
 12  Amount            11245 non-null  float64
dtypes: float64(5), object(8)
memory usage: 1.1+ MB
```

# let's start DATA CLEANING

In [6]:

```
#Deleting blank column
df.drop(['Zipcode'], axis=1, inplace=True)
```

In [7]:

```
#List of Columns Available
df.columns
```

Out[7]:

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Age', 'Age Group', 'Gender',
       'State', 'Zone', 'Profession', 'Product_Category', 'Orders', 'Amount'],
      dtype='object')
```

In [8]:

```
#check for null values
pd.isnull(df).sum()
```

Out[8]:

```
User_ID             12
Cust_name           12
Product_ID          12
Age                 12
Age Group           12
Gender              12
State               12
Zone                12
Profession          12
Product_Category    12
Orders              12
Amount              12
dtype: int64
```

In [9]:

```
# drop null values
df.dropna(how='all', inplace=True)
```

In [10]:

```
df.shape
```

Out[10]:

```
(11245, 12)
```

In [11]:
```
#replace valvue o genderr column
df['Gender'] = df['Gender'].replace('M', 'Male')
```

In [12]:
```
#view only male gender data
df[df['Gender'] == 'Male']
```

Out[12]:

| | User_ID | Cust_name | Product_ID | Age | Age Group | Gender | State | Zone |
|---|---|---|---|---|---|---|---|---|
| **3** | 1001425.0 | Virendra | P00237842 | 16.0 | 0-17 | Male | Karnataka | South |
| **4** | 1000588.0 | Vishal | P00057942 | 28.0 | 26-35 | Male | Gujarat | West |
| **5** | 1000588.0 | Suuraj | P00057942 | 28.0 | 26-35 | Male | Himachal Pradesh | Northern |
| **8** | 1003224.0 | Kushal | P00205642 | 35.0 | 26-35 | Male | Uttar Pradesh | Central |
| **11** | 1003829.0 | Harsh | P00200842 | 34.0 | 26-35 | Male | Delhi | Central |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **11249** | 1005446.0 | Sheetal | P00297742 | 53.0 | 51-55 | Male | Gujarat | West |
| **11250** | 1005446.0 | Sheetal | P00297742 | 53.0 | 51-55 | Male | Madhya Pradesh | Central |
| **11252** | 1000695.0 | Manning | P00296942 | 19.0 | 18-25 | Male | Maharashtra | West |
| **11253** | 1004089.0 | Reichenbach | P00171342 | 33.0 | 26-35 | Male | Haryana | Northern |
| **11255** | 1004023.0 | Noonan | P00059442 | 37.0 | 36-45 | Male | Karnataka | South |

3408 rows × 12 columns

In [13]:
```
#view only FEmale gender data
df[df['Gender'] == 'Female']
```

Out[13]:

| | User_ID | Cust_name | Product_ID | Age | Age Group | Gender | State | Zone |
|---|---|---|---|---|---|---|---|---|
| **0** | 1002903.0 | Anvi | P00125942 | 27.0 | 26-35 | Female | Maharashtra | Wes |
| **1** | 1000732.0 | Shanta | P00110942 | 34.0 | 26-35 | Female | Andhra Pradesh | South |
| **2** | 1001990.0 | Sheetal | P00118542 | 16.0 | 0-17 | Female | Uttar Pradesh | Centra |
| **6** | 1001132.0 | Sejal | P00018042 | 25.0 | 18-25 | Female | Uttar Pradesh | Centra |
| **7** | 1002092.0 | Shivangi | P00273442 | 61.0 | 55+ | Female | Maharashtra | Wes |
| **...** | ... | ... | ... | ... | ... | ... | ... | .. |
| **11247** | 1003032.0 | Matthias | P00058042 | 33.0 | 26-35 | Female | Delhi | Centra |
| **11248** | 1004344.0 | Hildebrand | P00185442 | 27.0 | 26-35 | Female | Delhi | Centra |
| **11251** | 1004140.0 | Bertelson | P00057442 | 31.0 | 26-35 | Female | Delhi | Centra |
| **11254** | 1001209.0 | Oshin | P00201342 | 40.0 | 36-45 | Female | Madhya Pradesh | Centra |
| **11256** | 1002744.0 | Brumley | P00281742 | 19.0 | 18-25 | Female | Maharashtra | Wes |

7837 rows × 12 columns

# NOW LET'S LEARN EDA - EXPLORATORY DATA ANALYSIS

In [14]:

```
# DESCRBE() METHOD RETURNS DESCRIPTION OF THE DATA IN THE DATA FRAME(i.e.count,s
df.describe()
```

Out[14]:

| | User_ID | Age | Orders | Amount |
|---|---|---|---|---|
| **count** | 1.124500e+04 | 11245.000000 | 11245.000000 | 11245.000000 |
| **mean** | 1.003004e+06 | 35.415651 | 3.500311 | 9461.934237 |
| **std** | 1.716207e+03 | 12.756369 | 1.713706 | 5234.426634 |
| **min** | 1.000001e+06 | 12.000000 | 1.000000 | 188.000000 |
| **25%** | 1.001492e+06 | 27.000000 | 2.000000 | 5443.000000 |
| **50%** | 1.003065e+06 | 33.000000 | 4.000000 | 8109.000000 |
| **75%** | 1.004429e+06 | 43.000000 | 5.000000 | 12683.000000 |
| **max** | 1.006040e+06 | 92.000000 | 6.000000 | 29350.000000 |

In [15]:

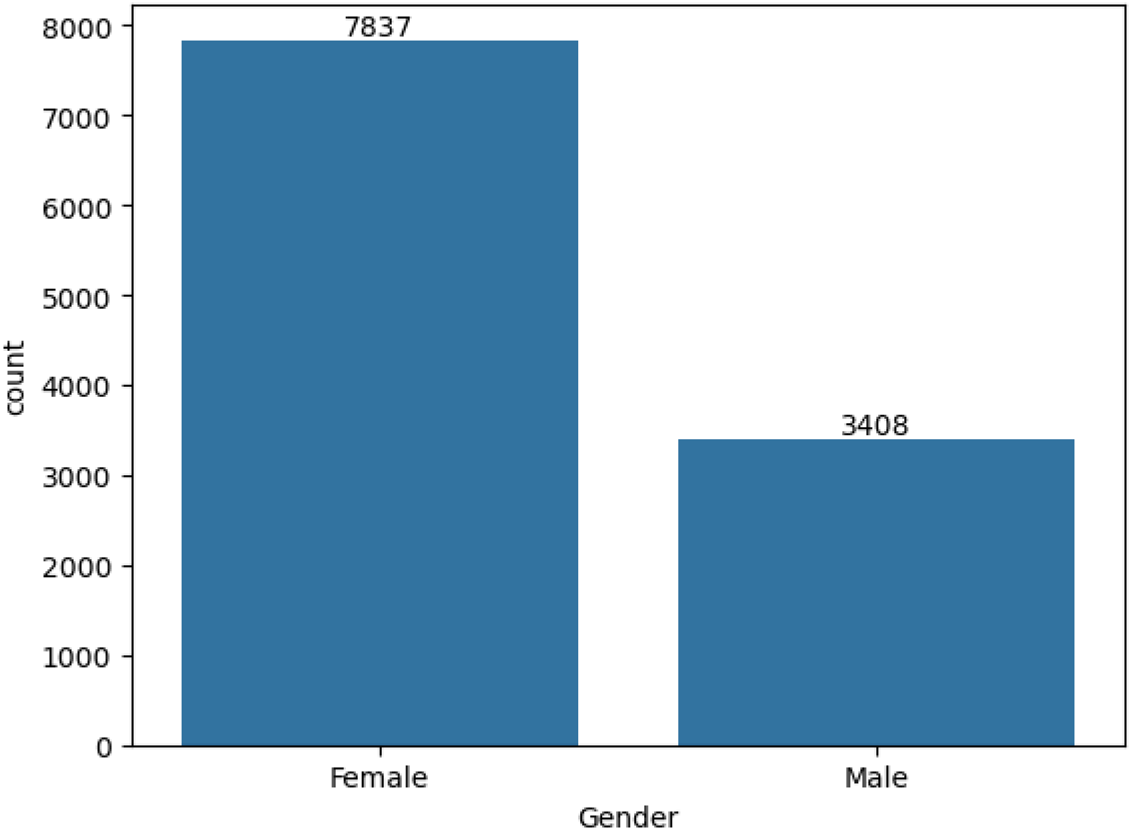
```
#use describe () for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

Out[15]:

| | Age | Orders | Amount |
|---|---|---|---|
| **count** | 11245.000000 | 11245.000000 | 11245.000000 |
| **mean** | 35.415651 | 3.500311 | 9461.934237 |
| **std** | 12.756369 | 1.713706 | 5234.426634 |
| **min** | 12.000000 | 1.000000 | 188.000000 |
| **25%** | 27.000000 | 2.000000 | 5443.000000 |
| **50%** | 33.000000 | 4.000000 | 8109.000000 |
| **75%** | 43.000000 | 5.000000 | 12683.000000 |
| **max** | 92.000000 | 6.000000 | 29350.000000 |

In [16]:

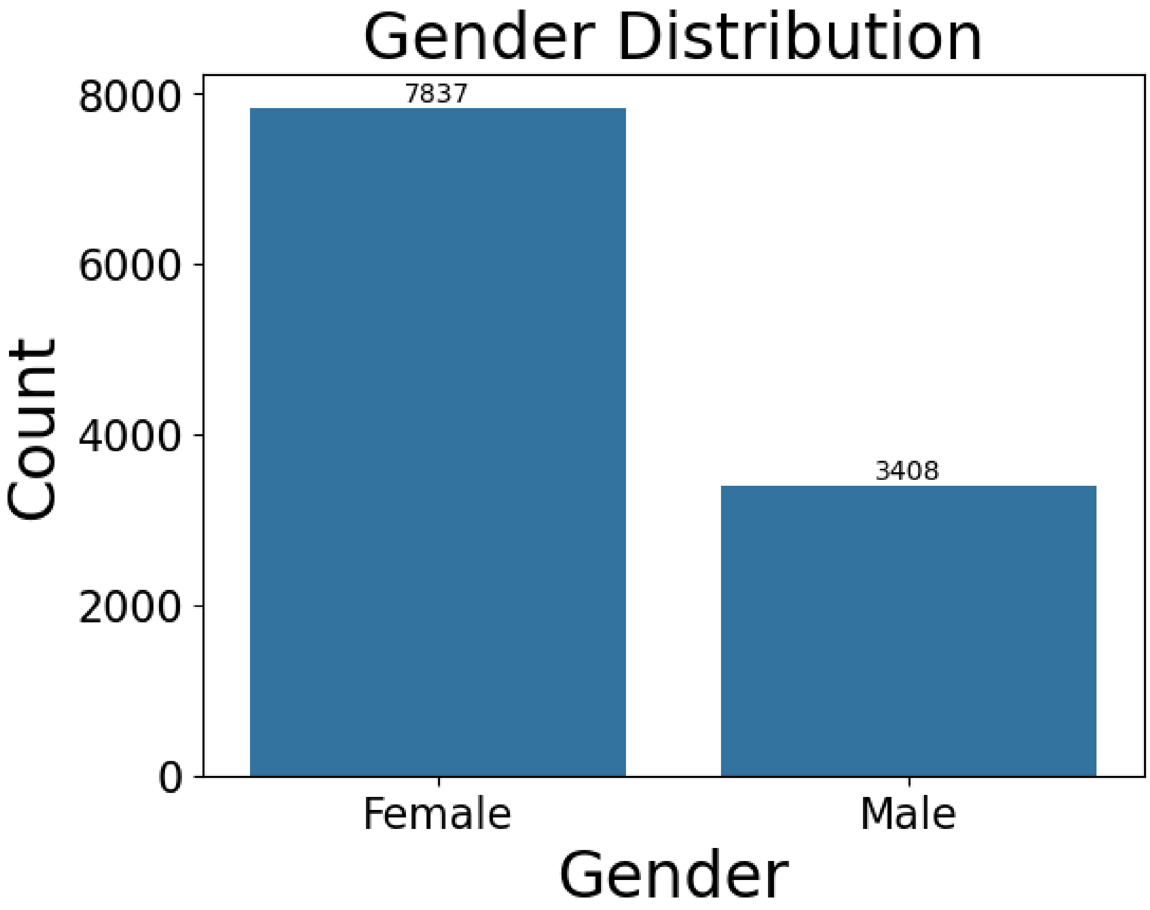
```
# total transaction count by gender wise in bar chart
ax=sns.countplot(x='Gender', data=df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



In [17]:

```
# Total Transactions count by Gender Wise in Bar Chart
ax = sns.countplot(x = 'Gender', data = df)

# Set Title and Labels with Font Size
for bars in ax.containers:
    ax.bar_label(bars)
    ax.set_title('Gender Distribution', fontsize=24)
    ax.set_xlabel('Gender', fontsize=24)
    ax.set_ylabel('Count', fontsize=24)
    ax.tick_params(axis='both', labelsize=16)
plt.show()
```



In [18]:

```
# 2 . Gender wise distribution in Pie Chart
```

In [19]:

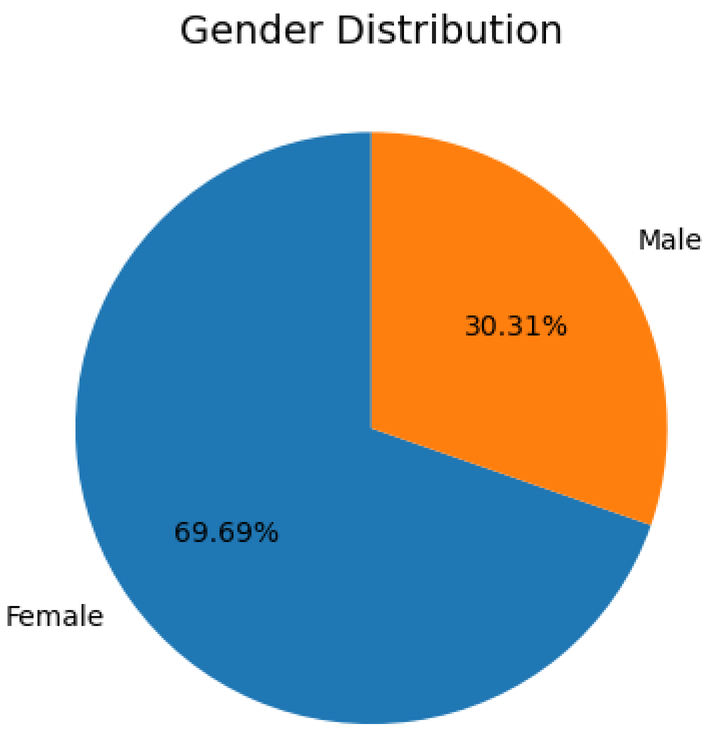
```
# total transaction count by gender wise
gender_count = df['Gender'].value_counts()
print(gender_count)
```

```
Gender
Female    7837
Male      3408
Name: count, dtype: int64
```

In [20]:

```
# total transaction count by gender wise pie chart
gender_count = df['Gender'].value_counts()

plt.pie(
    gender_count,
    labels=gender_count.index,
    autopct='%.2f%%',
    startangle=90,
)
# add a title
plt.title('Gender Distribution', fontsize=14)
plt.show()
```



In [21]:

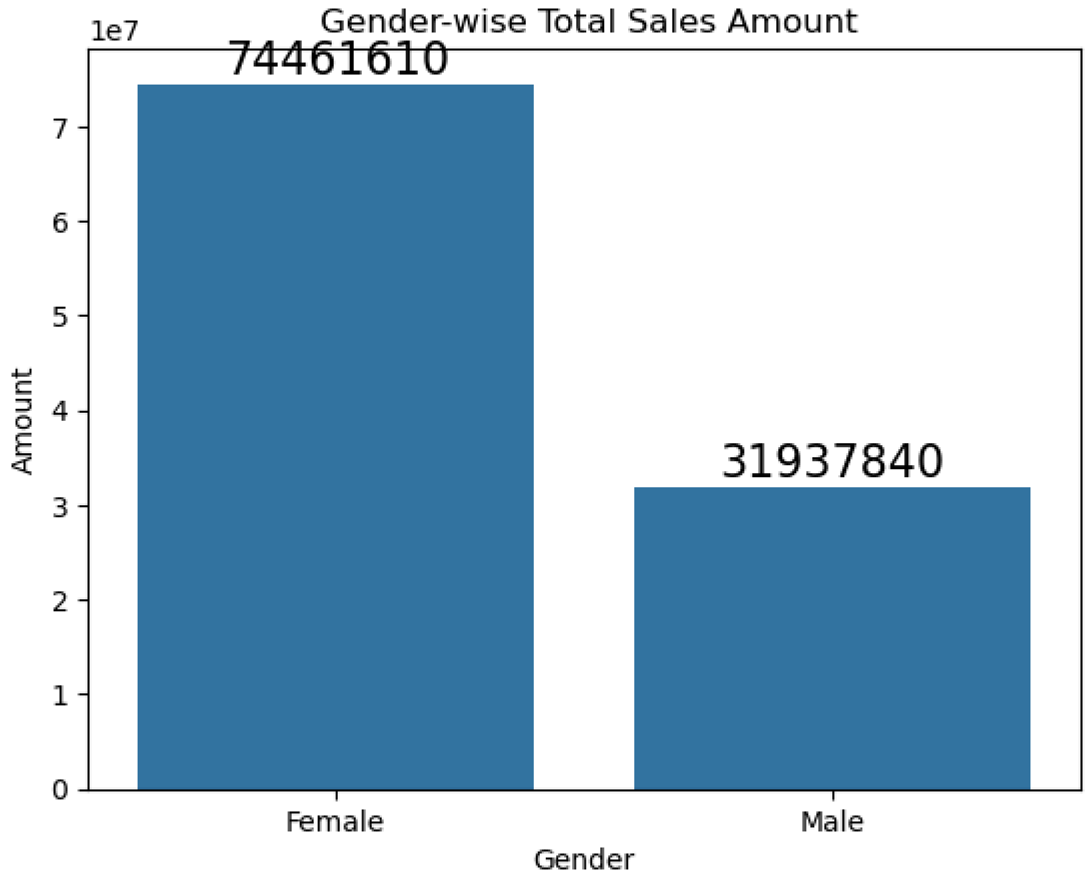
```
df.groupby('Gender', as_index=False)['Amount'].sum()
```

Out[21]:

|   | Gender | Amount |
|---|---|---|
| **0** | Female | 74461610.49 |
| **1** | Male | 31937840.00 |

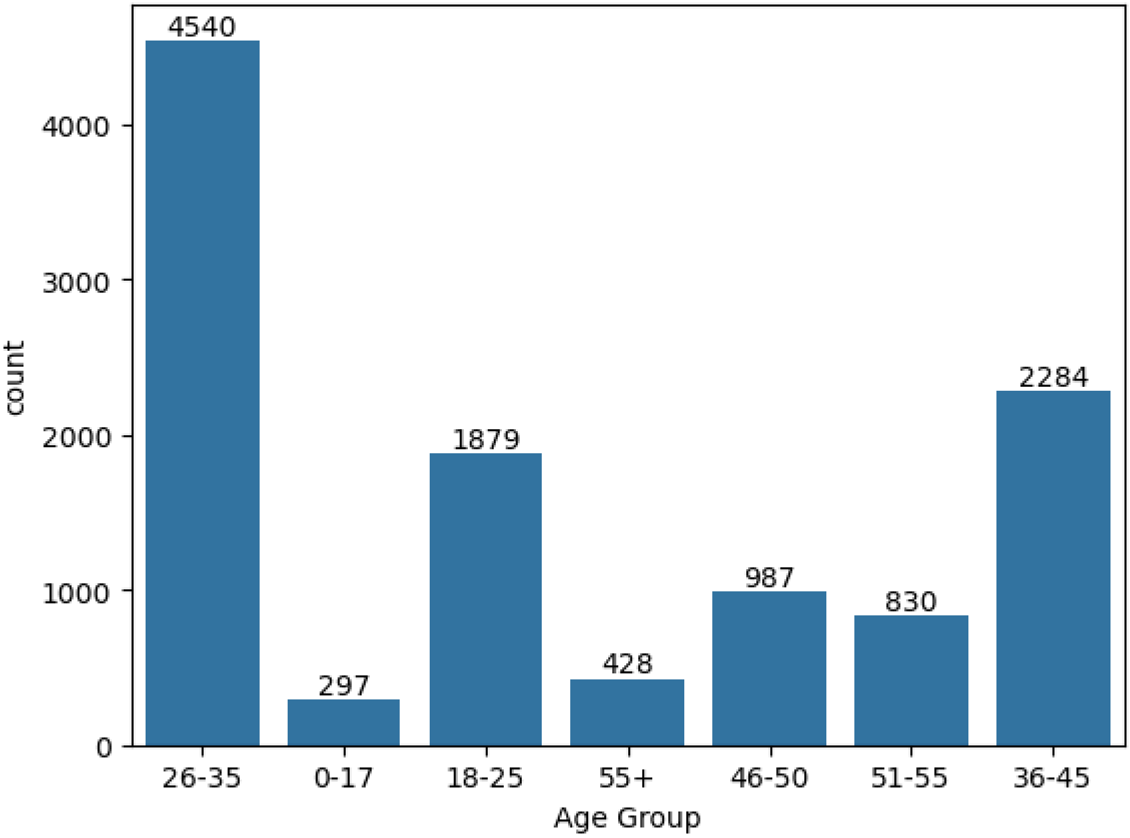
In [22]:

```
# gender wise total sales amount
# step 1: group data by gender and the sum the amount
Gen_Wise_Sales = df.groupby('Gender', as_index=False)['Amount'].sum().sort_value
# create the bar plot
ax = sns.barplot(x='Gender', y='Amount', data=Gen_Wise_Sales)
# step 3 : add data labels on top of bars
for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=16, color='black')
#step 4 : add titles
plt.title(' Gender-wise Total Sales Amount', fontsize=12)
plt.show()
```



## age group wise total transactions

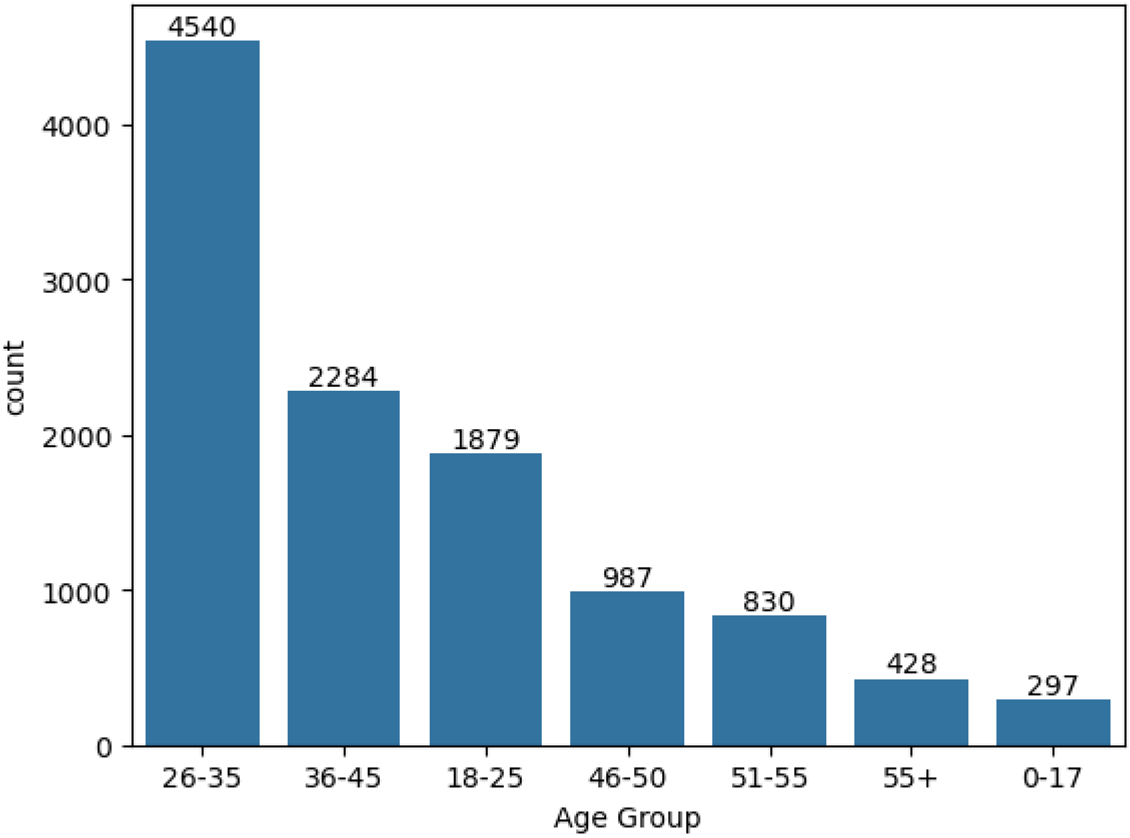
In [23]:
```
# age group wise transactions count in bar chart
ax = sns.countplot(data = df, x='Age Group')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```



In [24]:

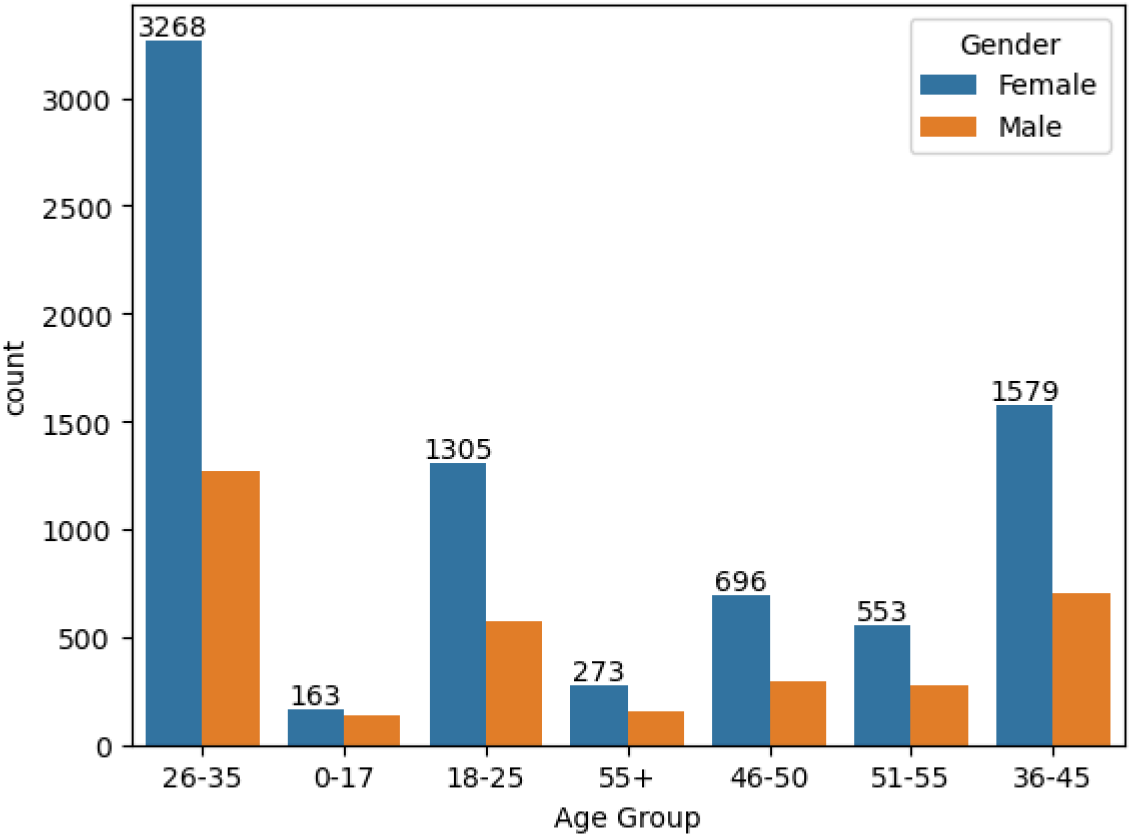
```
# s-1 ae group wise transactions count( sorting)
age_grup_count=df['Age Group'].value_counts().sort_values(ascending= False)
print(age_grup_count)
# s-2 use ordered category for sorting in countplot
sns_order = age_grup_count.index
ax = sns.countplot(data = df, x='Age Group', order=sns_order)
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```

```
Age Group
26-35    4540
36-45    2284
18-25    1879
46-50     987
51-55     830
55+       428
0-17      297
Name: count, dtype: int64
```



In [25]:

```
# age group wise transactions distribution
ax = sns.countplot(data = df, x='Age Group', hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```

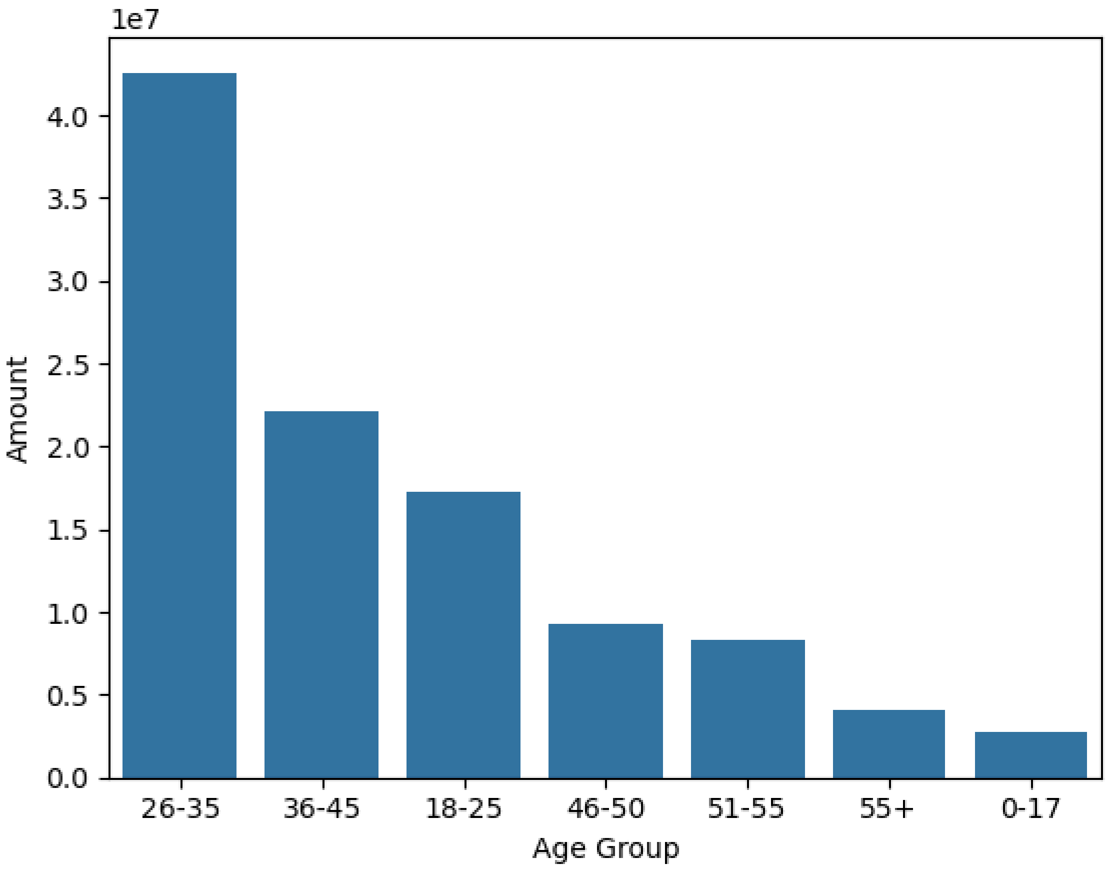


# from aove graphs we can see that most of byers are of age group between 26-35yrs female

In [26]:

```
# age group wise total amount
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_value

sns.barplot(x = 'Age Group',y= 'Amount', data= sales_age)
plt.show()
```

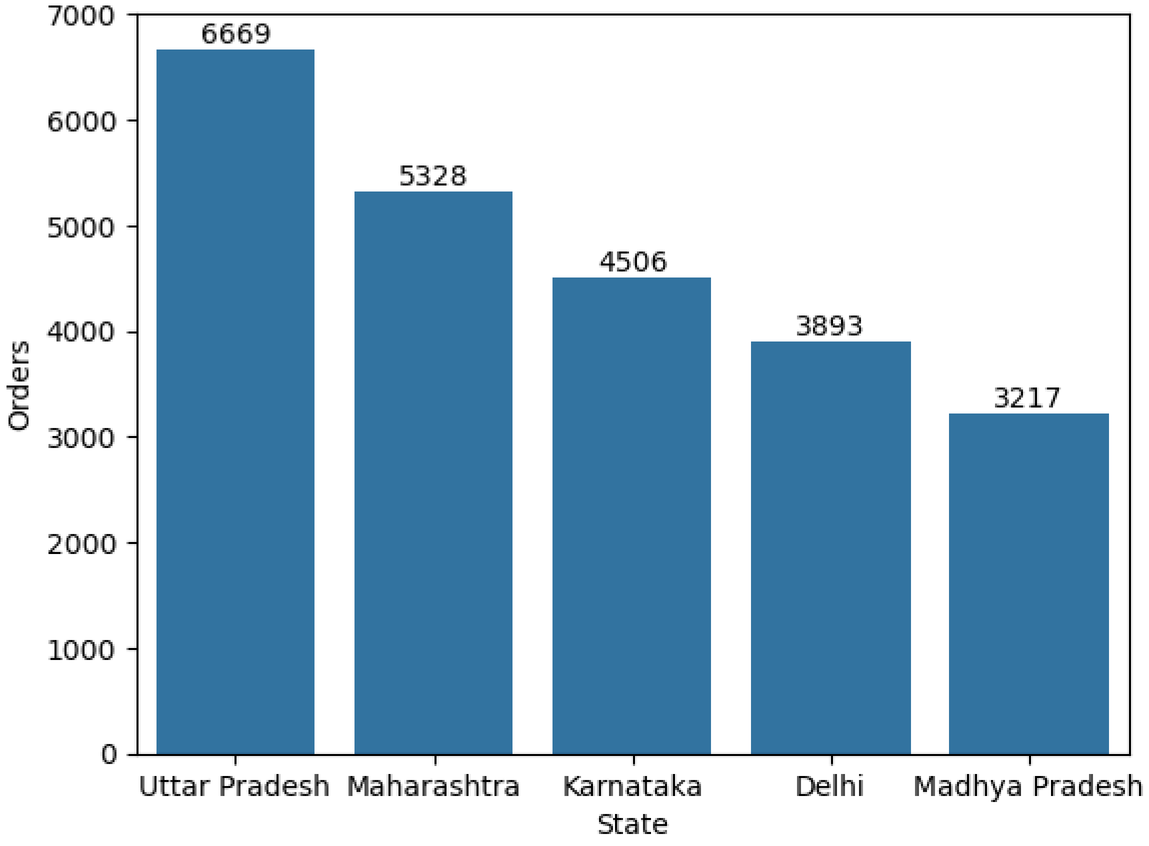


# state wise anlysis

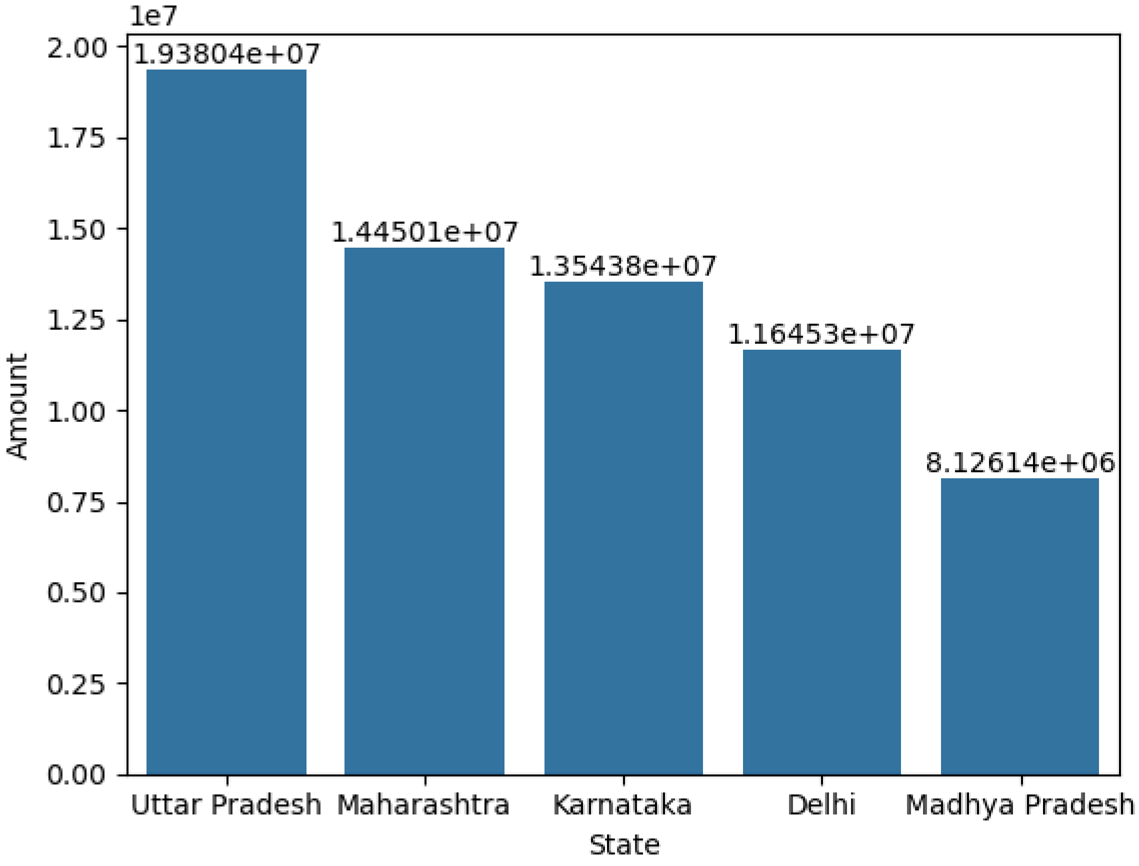
In [32]:

```
# order wise top 5 state
order_state = df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(b

 #sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data = order_state, x = 'State',y= 'Orders')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```



In [35]:
```
#amount wisr top 5 state
sales_sate =df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by=
#sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data = sales_sate, x = 'State',y= 'Amount')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```

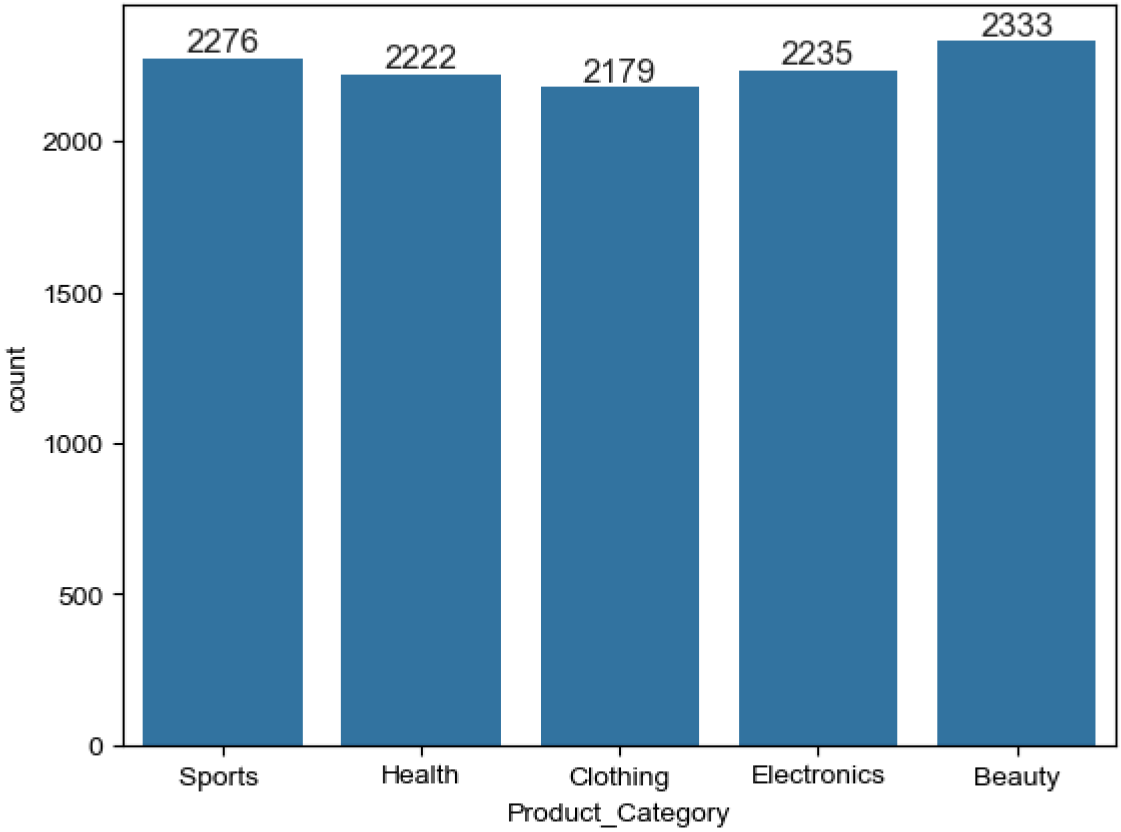


## Product_Category Analysis

In [41]:

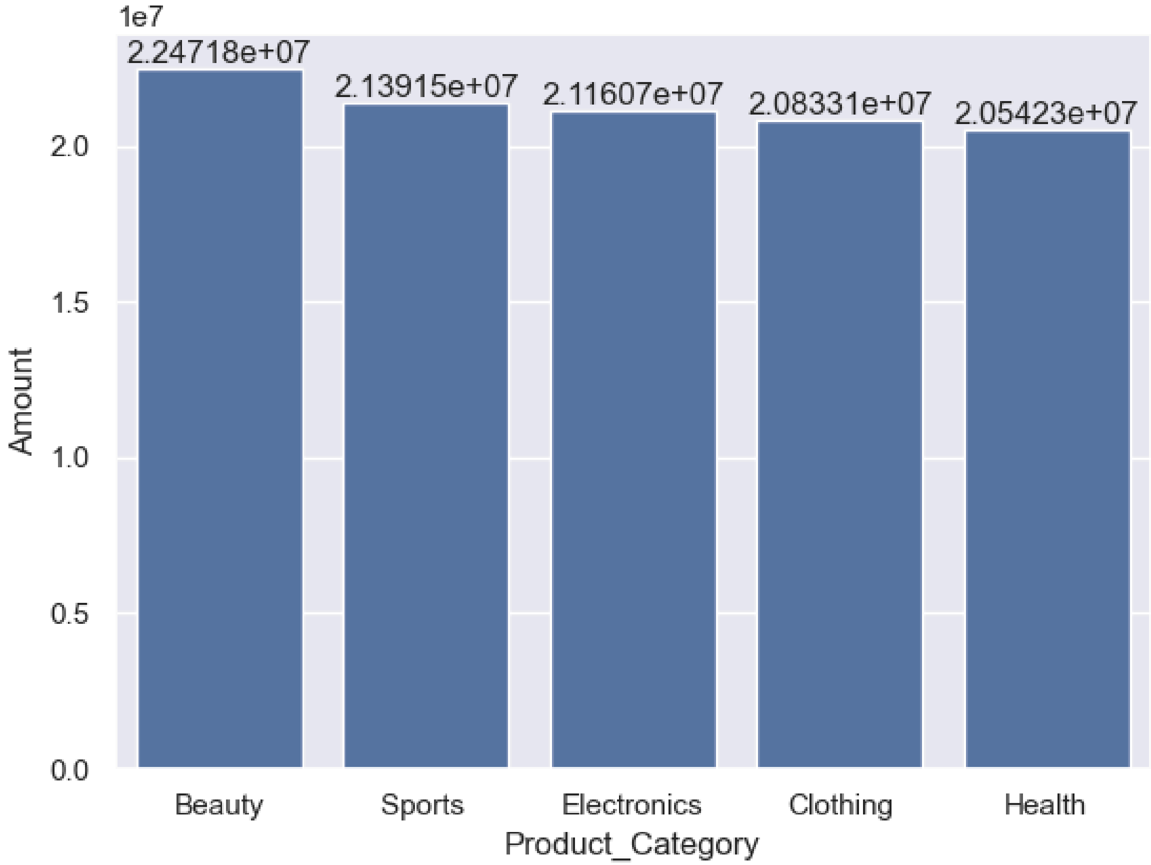
```
# product category wise transaction count
ax = sns.countplot(data=df,x='Product_Category')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



In [42]:

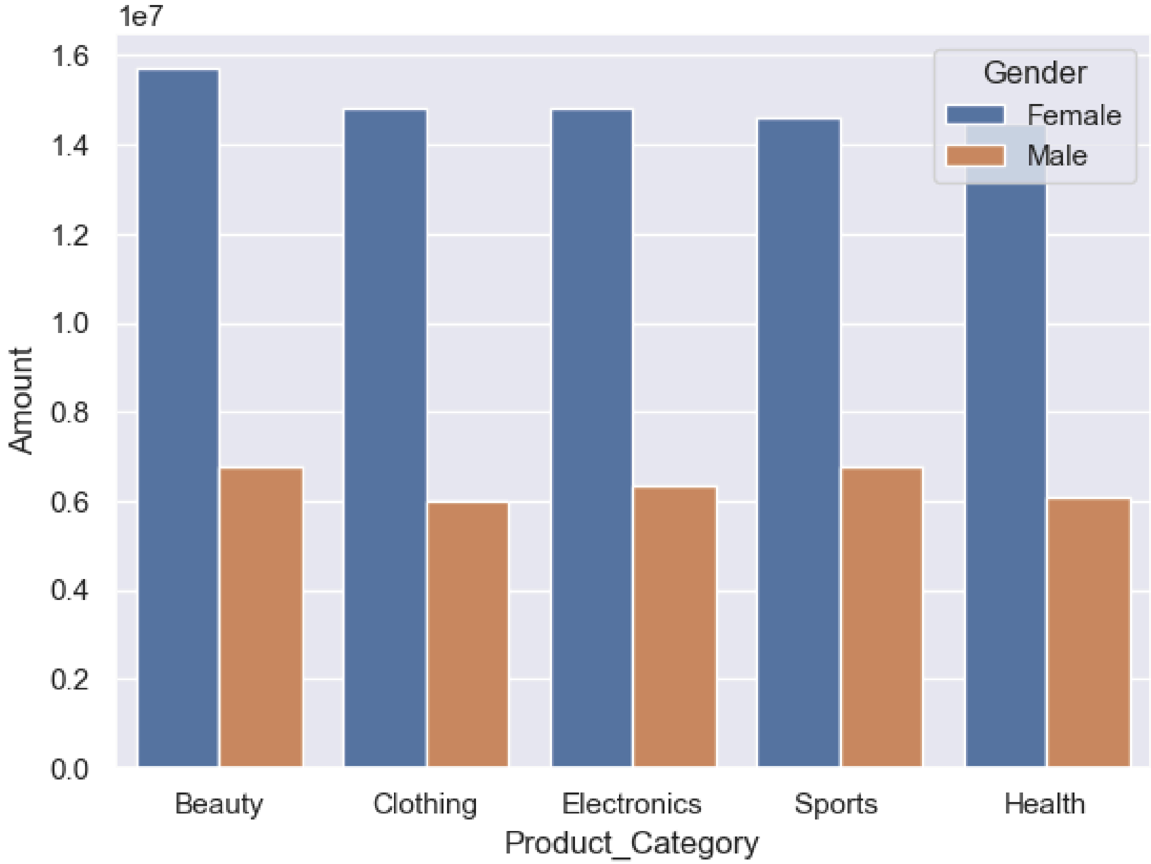
```
# amount wise prduct category in bar chart
sales_state =df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sor
#sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```



In [48]:

```
# product category and gender wise transactions
sales_pro =df.groupby(['Product_Category','Gender'],as_index=False)['Amount'].su
sns.barplot(data = sales_pro, x = 'Product_Category',y= 'Amount', hue='Gender')

plt.show()
```



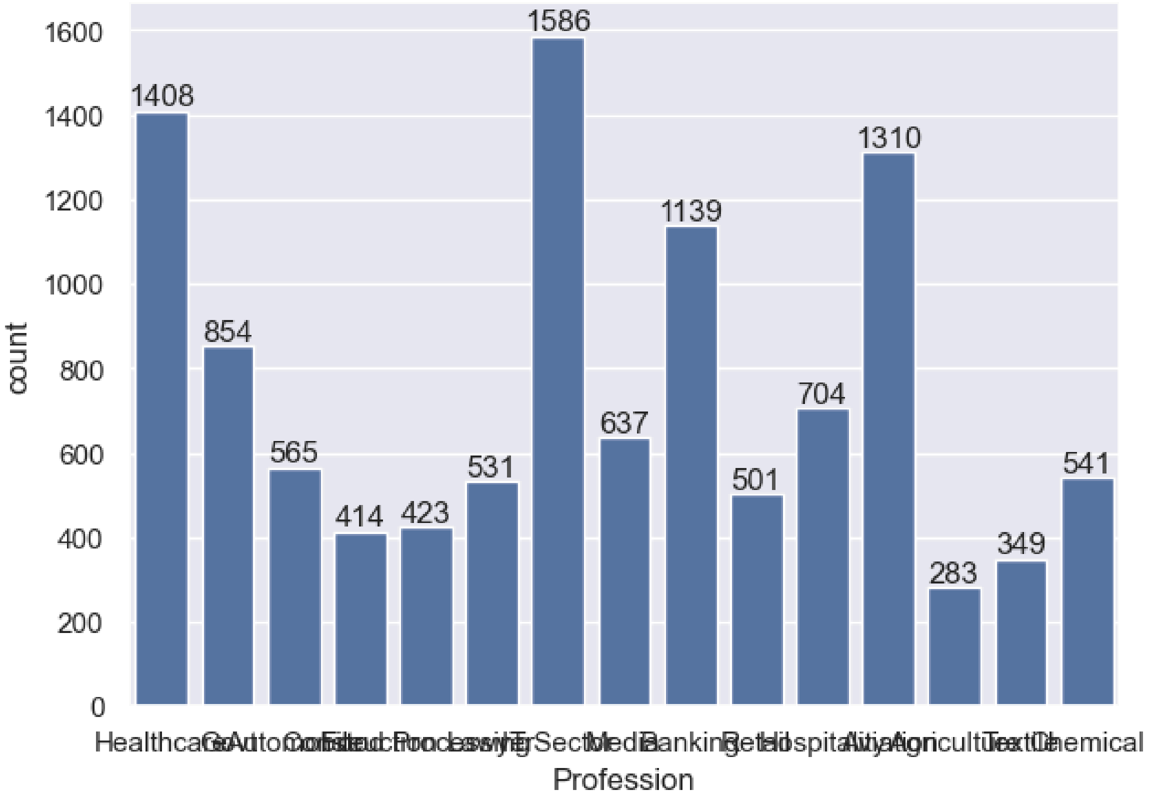
In [ ]:

```
* from the above graphs we can see the most byers are female
```

# Profession wise Analysis

In [55]:

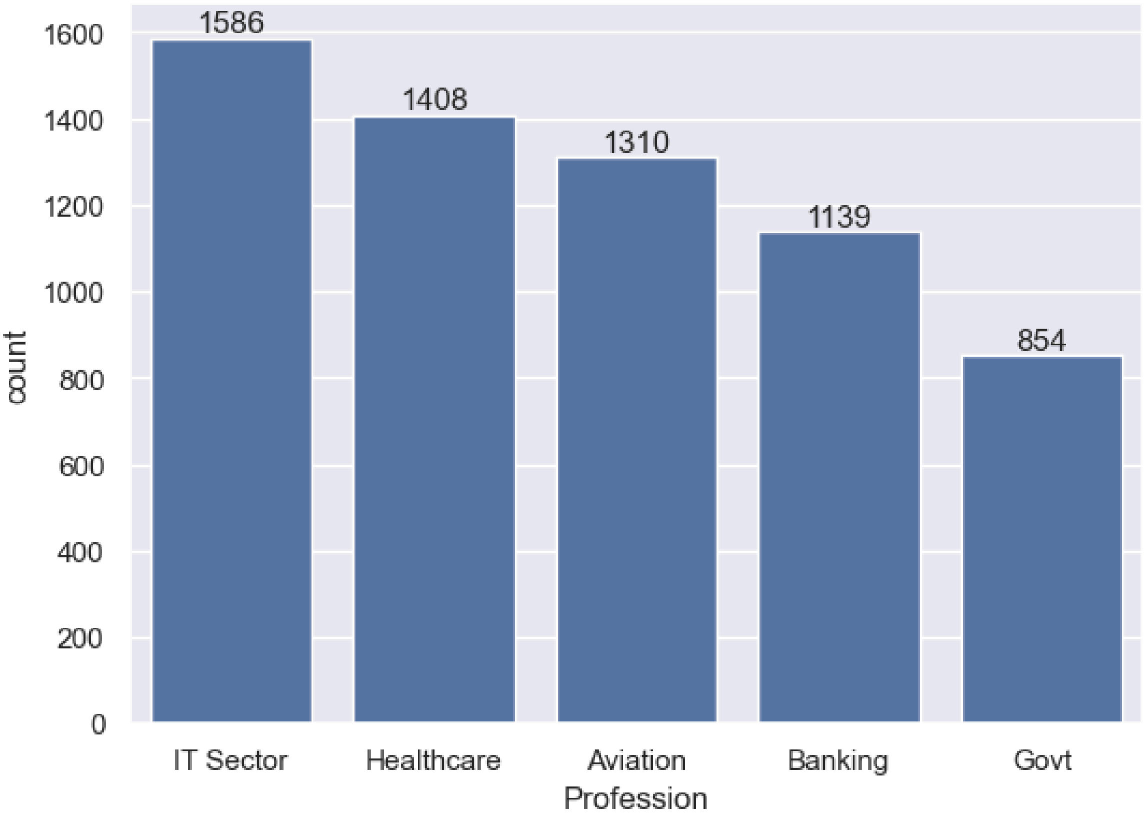
```
# Profession wise Analysis
ax=sns.countplot(data=df, x='Profession')
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```



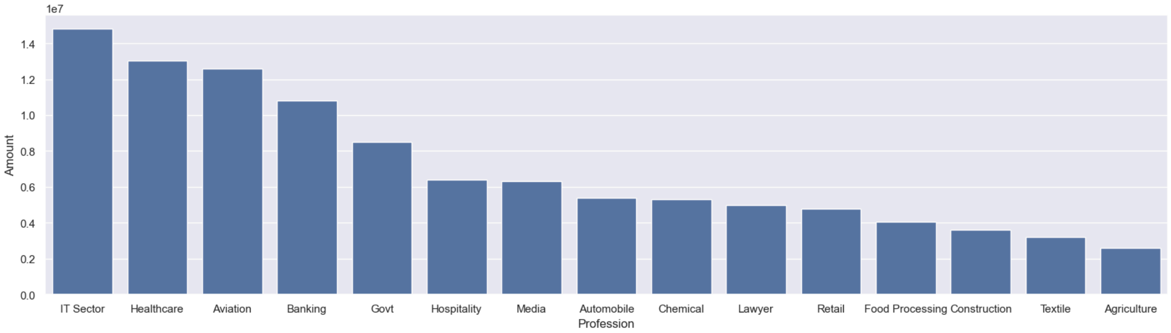
In [56]:

```
# s-1 ae group wise transactions count( sorting)
profession_count=df['Profession'].value_counts().sort_values(ascending= False).h
print(profession_count)
# s-2 use ordered category for sorting in countplot
sns_order = profession_count.index
ax = sns.countplot(data = df, x='Profession', order=sns_order)
for bars in ax.containers:
    ax.bar_label(bars)
    plt.show()
```

```
Profession
IT Sector     1586
Healthcare    1408
Aviation      1310
Banking       1139
Govt           854
Name: count, dtype: int64
```

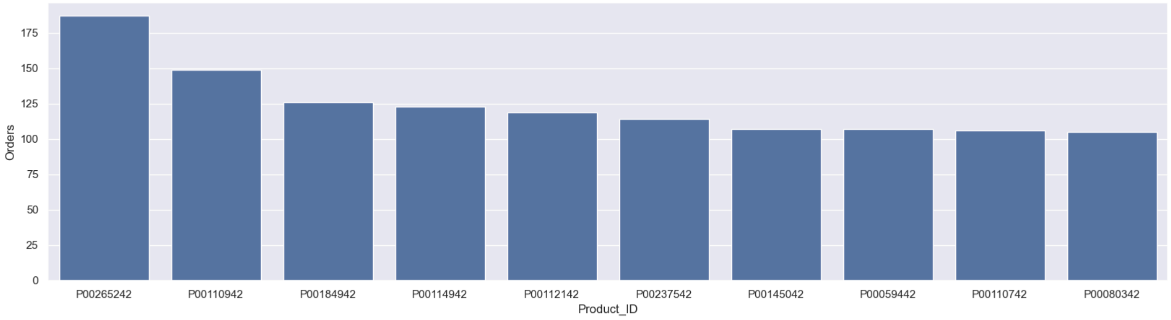


In [60]:
```
sales_state_1 =df.groupby(['Profession'],as_index=False)['Amount'].sum().sort_va
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.barplot(data =sales_state_1, x ='Profession',y= 'Amount')
plt.show()
```



In [ ]:
```
# fro above graphs we see that most byers are working from it, healthcare, and a
```

In [61]:
```
# orders wise top 10 products
sales_state_2 =df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_va
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.barplot(data =sales_state_2, x ='Product_ID',y= 'Orders')
plt.show()
```



# CONCLUSION

*** FEMALE AGE GROUP 16-35YRS FROM UP, MAHARASTRA AND KARNATAKA WORKING IN IT, HEALTHCARE AND AVIATION ARE MORE LIKELY TO BUY PRODUCT FROM BEAUTY , SPORTS AND ELECTRONICS