In [2]:

```python
# importing pyhton libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [5]:

```python
# importing csv file
df=pd.read_csv('Diwali Sales Data.csv', encoding='unicode_escape')
```

In [6]:

```python
df
```

Out[6]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2
...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3

11251 rows × 15 columns

In [8]:

```python
df.shape
```

Out[8]:

```
(11251, 15)
```

In [9]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11251 non-null  int64  
 1   Cust_name         11251 non-null  object 
 2   Product_ID        11251 non-null  object 
 3   Gender            11251 non-null  object 
 4   Age Group         11251 non-null  object 
 5   Age               11251 non-null  int64  
 6   Marital_Status    11251 non-null  int64  
 7   State             11251 non-null  object 
 8   Zone              11251 non-null  object 
 9   Occupation        11251 non-null  object 
 10  Product_Category  11251 non-null  object 
 11  Orders            11251 non-null  int64  
 12  Amount            11239 non-null  float64
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

In [12]:

```python
df.drop(['Status','unnamed1'],axis=1,inplace=True)  #dropped empty/unrelated columns
```

In [16]:

```python
df.isnull().sum().sum()                             #checking the null values in dataset
```

Out[16]:

```
12
```

In [17]:

```
df.dropna(inplace=True)                                    #dropping the null values
```

In [18]:

```
df.isna().sum().sum()                                      #checking the null values after dropping
```

Out[18]:

```
0
```

In [20]:

```
df.columns
```

Out[20]:

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

In [23]:

```
df.duplicated().sum()                                     #checking the duplicated values
```

Out[23]:

```
8
```

In [28]:

```
df.drop_duplicates(inplace=True)                          #dropping the duplicated values
```

In [31]:

```
df.describe().T                                            #for statistical data
```

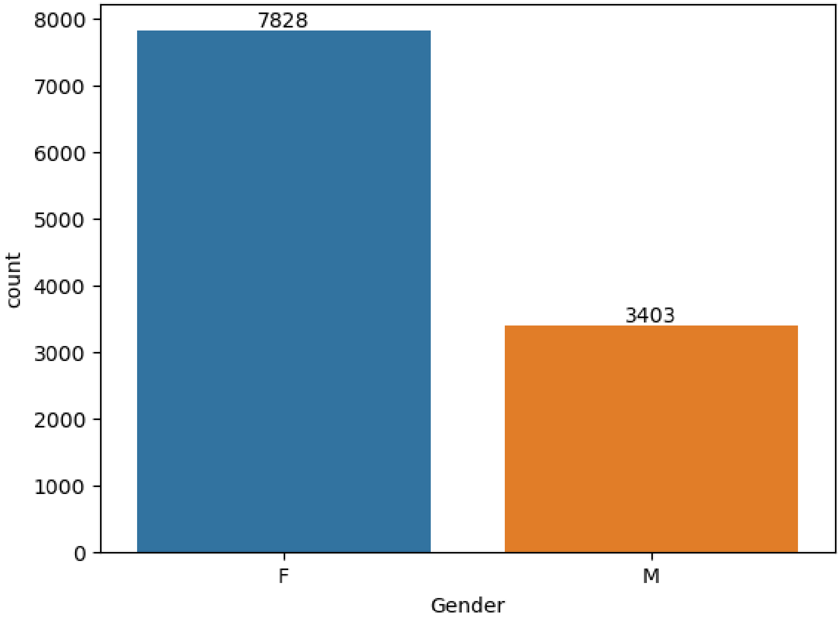
Out[31]:

	count	mean	std	min	25%	50%	75%	max
User_ID	11231.0	1.003004e+06	1716.054735	1000001.0	1001492.0	1003065.0	1004428.0	1006040.0
Age	11231.0	3.541198e+01	12.756116	12.0	27.0	33.0	43.0	92.0
Marital_Status	11231.0	4.199982e-01	0.493580	0.0	0.0	0.0	1.0	1.0
Orders	11231.0	2.489093e+00	1.114880	1.0	2.0	2.0	3.0	4.0
Amount	11231.0	9.454085e+03	5221.728776	188.0	5443.0	8109.0	12677.5	23952.0

Exploratory Data Analysis

In [32]:

```
# plotting a chart for gender and its count
ax=sns.countplot(x='Gender',data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```

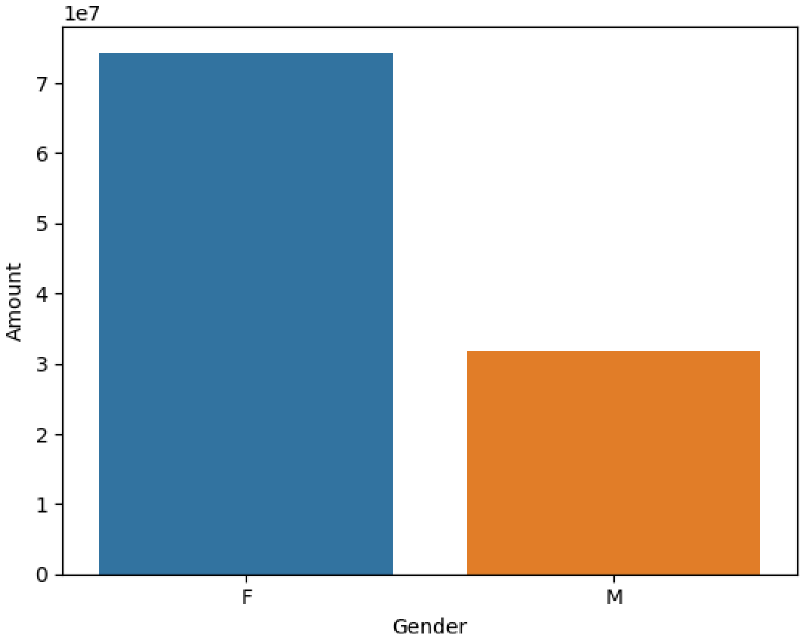


In [34]:

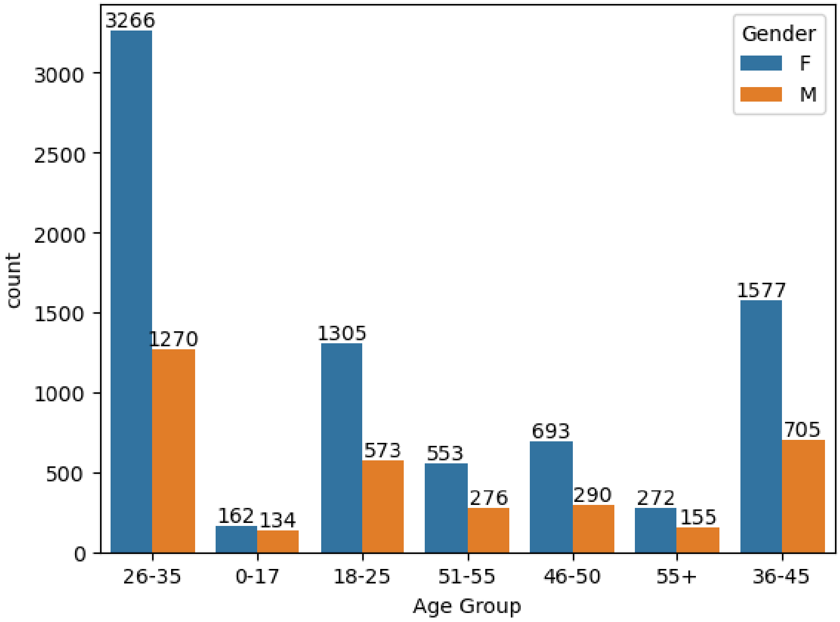
```
# plotting a bar chart for gender vs total amount
sales_gender = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gender)
```

Out[34]:

```
<Axes: xlabel='Gender', ylabel='Amount'>
```

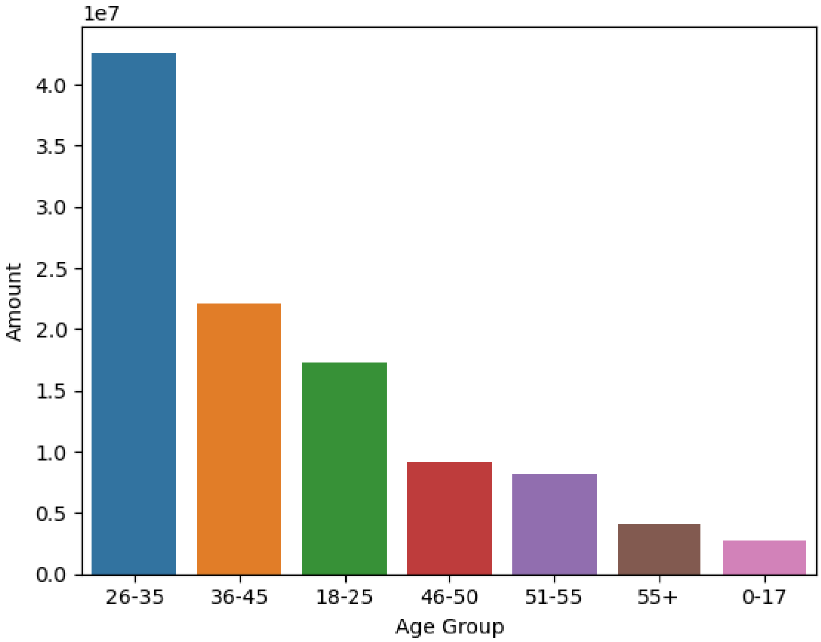


```
In [36]: # count of gender within respective age group
         ax=sns.countplot(data=df, x='Age Group', hue='Gender')
         for bars in ax.containers:
             ax.bar_label(bars)
```



```
In [38]: # Total amount vs Age Group
         amount_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
         sns.barplot(data=amount_age,x='Age Group',y='Amount')
```

```
Out[38]: <Axes: xlabel='Age Group', ylabel='Amount'>
```

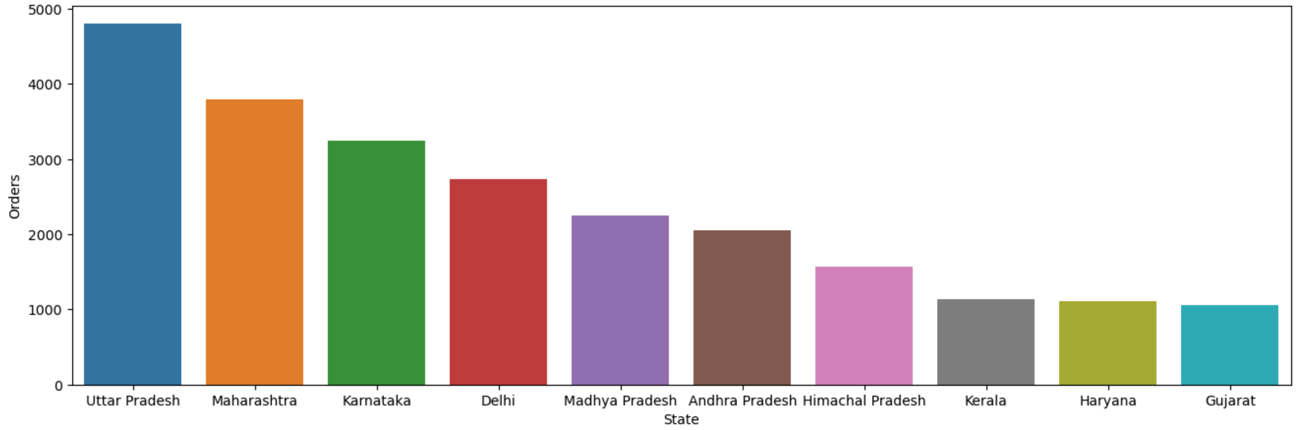


In [42]:

```
# Top 10 states having maximum number of orders
plt.figure(figsize=(16,5))
sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
sns.barplot(data=sales_state,x='State',y='Orders')
```

Out[42]:

```
<Axes: xlabel='State', ylabel='Orders'>
```

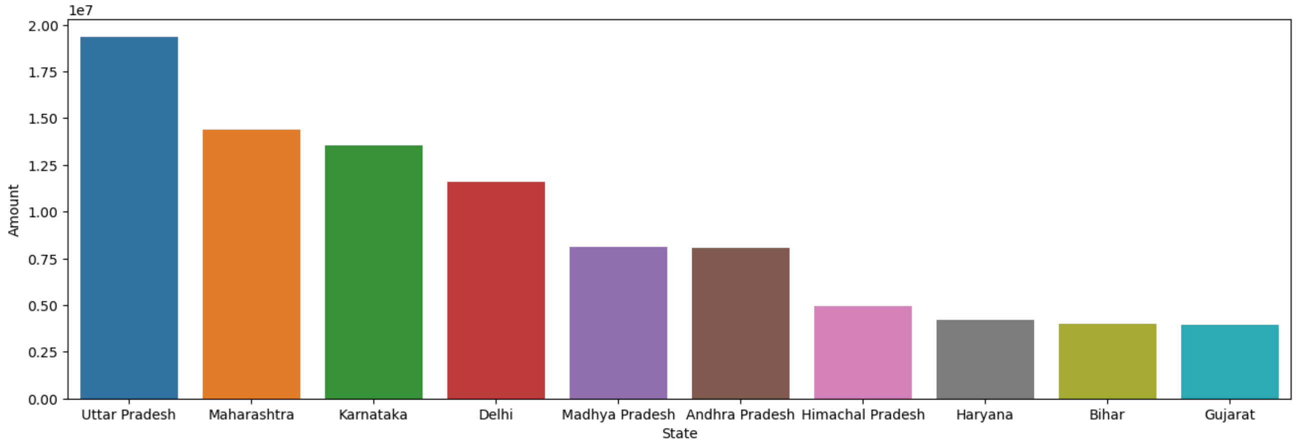


In [45]:

```
# Top 10 states having maximum amount of sales
plt.figure(figsize=(16,5))
amount_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)
sns.barplot(data=amount_state, x='State',y='Amount')
```

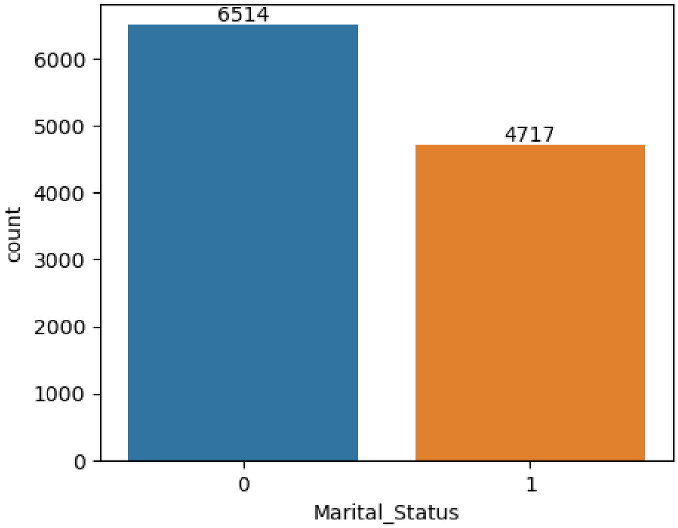
Out[45]:

```
<Axes: xlabel='State', ylabel='Amount'>
```



In [63]:

```
# Marital Status
plt.figure(figsize=(5,4))
ax = sns.countplot(data = df, x = 'Marital_Status')
for bars in ax.containers:
    ax.bar_label(bars)
```

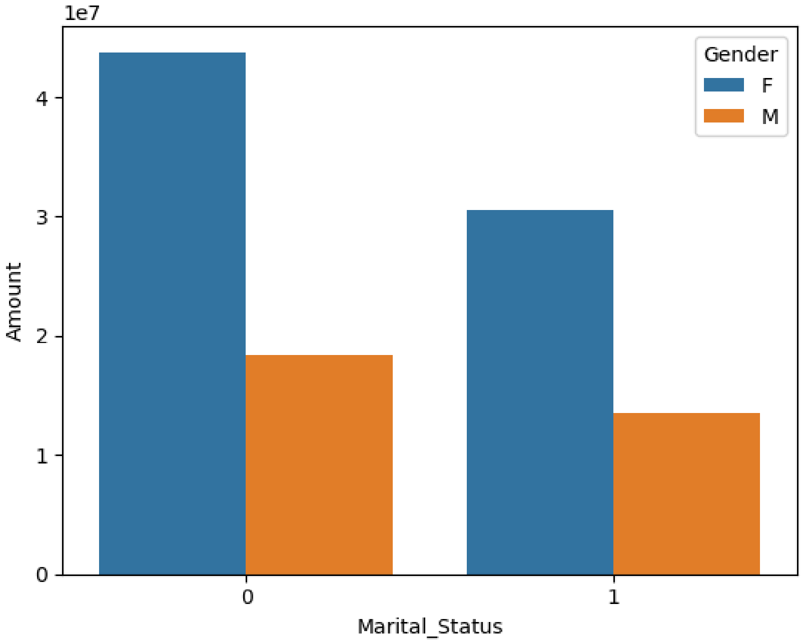


In [65]:

```
# Comparison of amount spending by Married and unmarried person
amount_marr_gender=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascendin
sns.barplot(data=amount_marr_gender,x='Marital_Status',y='Amount',hue='Gender')
```

Out[65]:

```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```

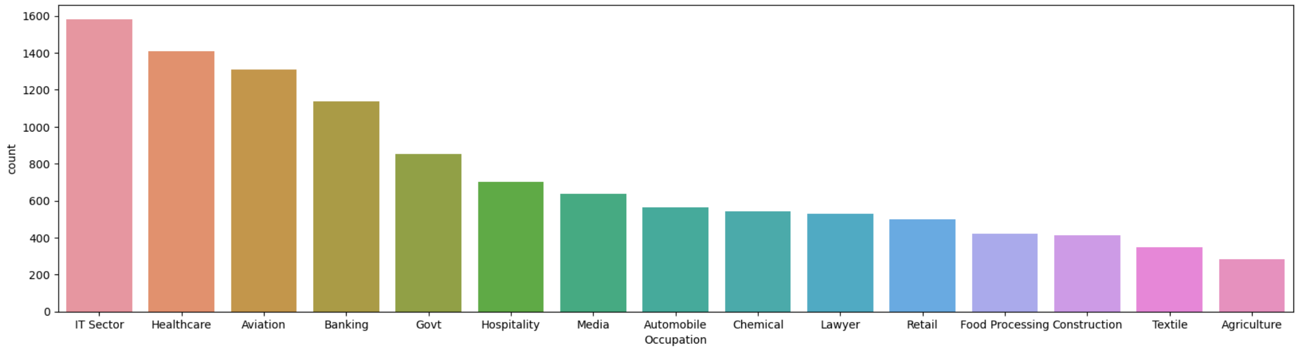


In [75]:

```
# Finding top occupations of people based on count of orders
plt.figure(figsize=(20,5))
sns.countplot(x='Occupation',data=df,order=df['Occupation'].value_counts().index)
```

Out[75]:

```
<Axes: xlabel='Occupation', ylabel='count'>
```

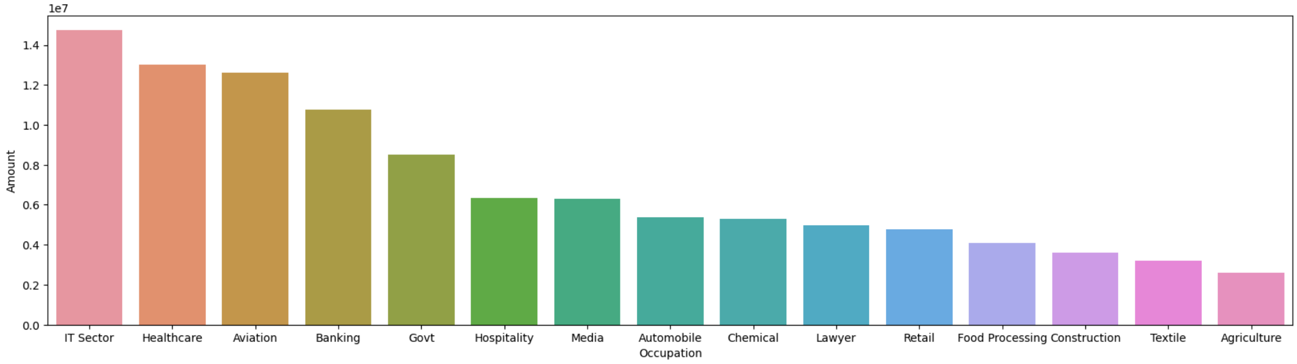


In [78]:

```
# Finding the top occupations based on amount spending
plt.figure(figsize=(20,5))
amount_occ=df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(data=amount_occ,x='Occupation',y='Amount')
```

Out[78]:

```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

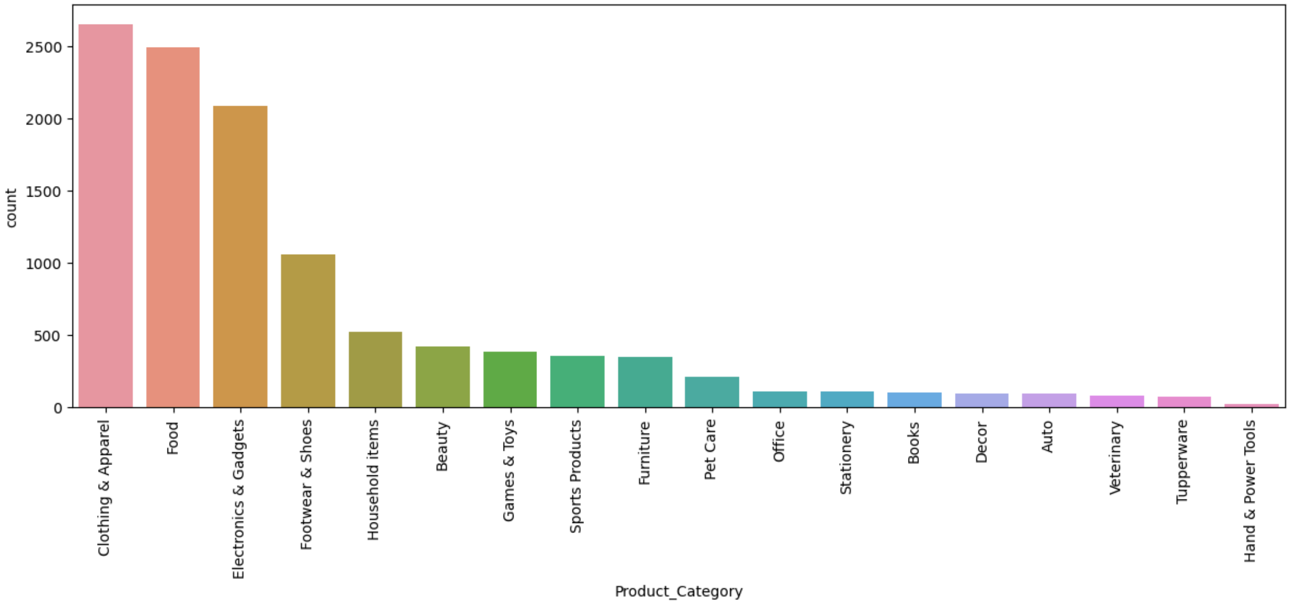


In [100…

```
# Number of orders based on Product Category
plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
sns.countplot(x='Product_Category',data=df,order=df['Product_Category'].value_counts().index)
```

Out[100]:

```
<Axes: xlabel='Product_Category', ylabel='count'>
```

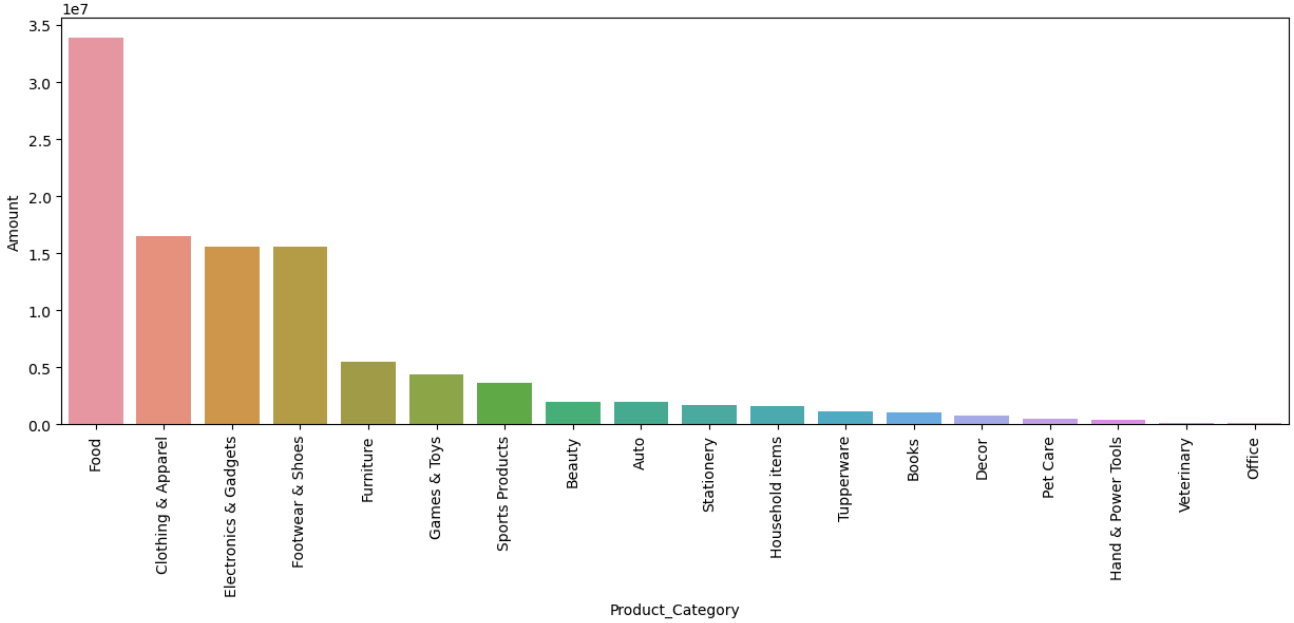


In [101…

```
# comparsion of product category based on sales amount
plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
product_amount=df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(data=product_amount,x='Product_Category',y='Amount')
```

Out[101]:

```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```

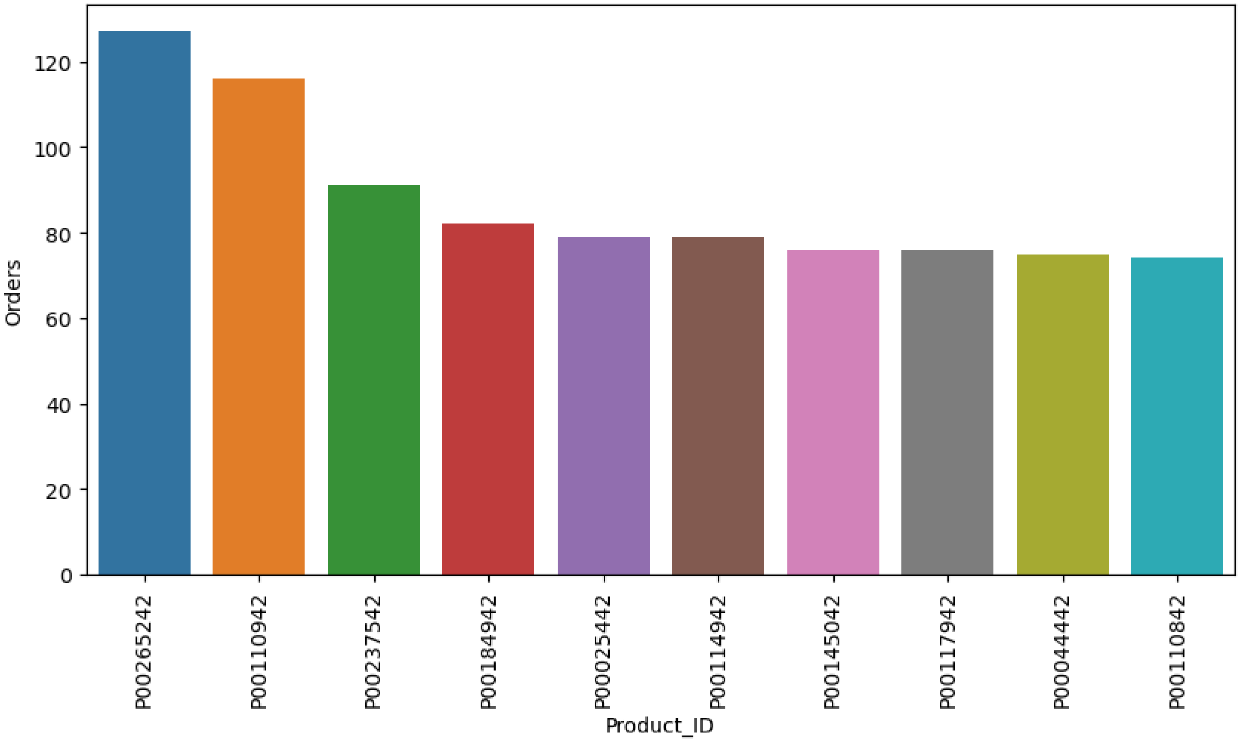


In [105…

```
# Top selling products
plt.figure(figsize=(10,5))
plt.xticks(rotation=90)
product_orders=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
sns.barplot(data=product_orders,x='Product_ID',y='Orders')
```

Out[105]:

```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



conclusion

From the above EDA on the provided dataset, we found that:

1. Most of the buyers are females and even the purchasing power of females are greater than men
2. Most of the buyers are of age group between 26-35 yrs Female
3. Most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively
4. Most of the buyers are married (women) and they have high purchasing power
5. Most of the buyers are working in IT, Healthcare and Aviation sector

6. Most of the sold products are from Food, Clothing and Electronics category

Married women in age group (26-35) Years from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category.