In [1]:
```
# import python libraries

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

In [4]:
```
# import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

In [6]:
```
df.shape
```

Out[6]:
```
(11251, 15)
```

In [8]:
```
df.head(5)
```

Out[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	C
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	C
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	

In [9]:
```
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 [10]:
```
#drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

In [11]:
```
#check for null values
pd.isnull(df).sum()
```

Out[11]:
```
User_ID             0
Cust_name           0
Product_ID          0
Gender              0
Age Group           0
Age                 0
Marital_Status      0
State               0
Zone                0
Occupation          0
Product_Category    0
Orders              0
Amount             12
dtype: int64
```

In [13]:
```
# drop null values
df.dropna(inplace=True)
```

In [14]:
```
df.shape
```

Out[14]:
```
(11239, 13)
```

In [15]:
```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

In [16]:
```
df['Amount'].dtypes
```

Out[16]:
```
dtype('int32')
```

In [17]:
```
df.columns
```

Out[17]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [18]:
```
df.rename(columns= {'Marital_Status':'Shaadi'})
```

Out[18]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Zone	Oc
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	H
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Au
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Co
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	F
...	...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	A
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	H

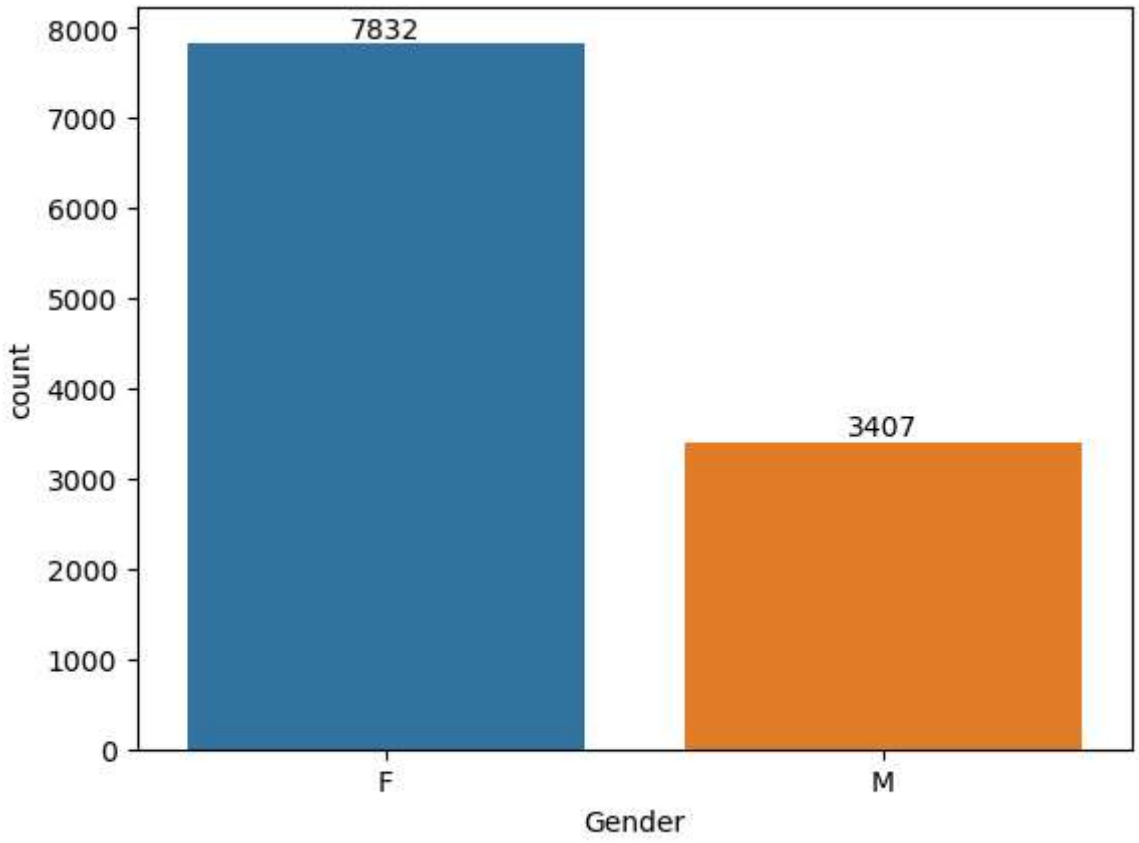
11239 rows × 13 columns

In [20]:
```
df[["Age", "Amount", "Orders"]].describe()
```

Out[20]:

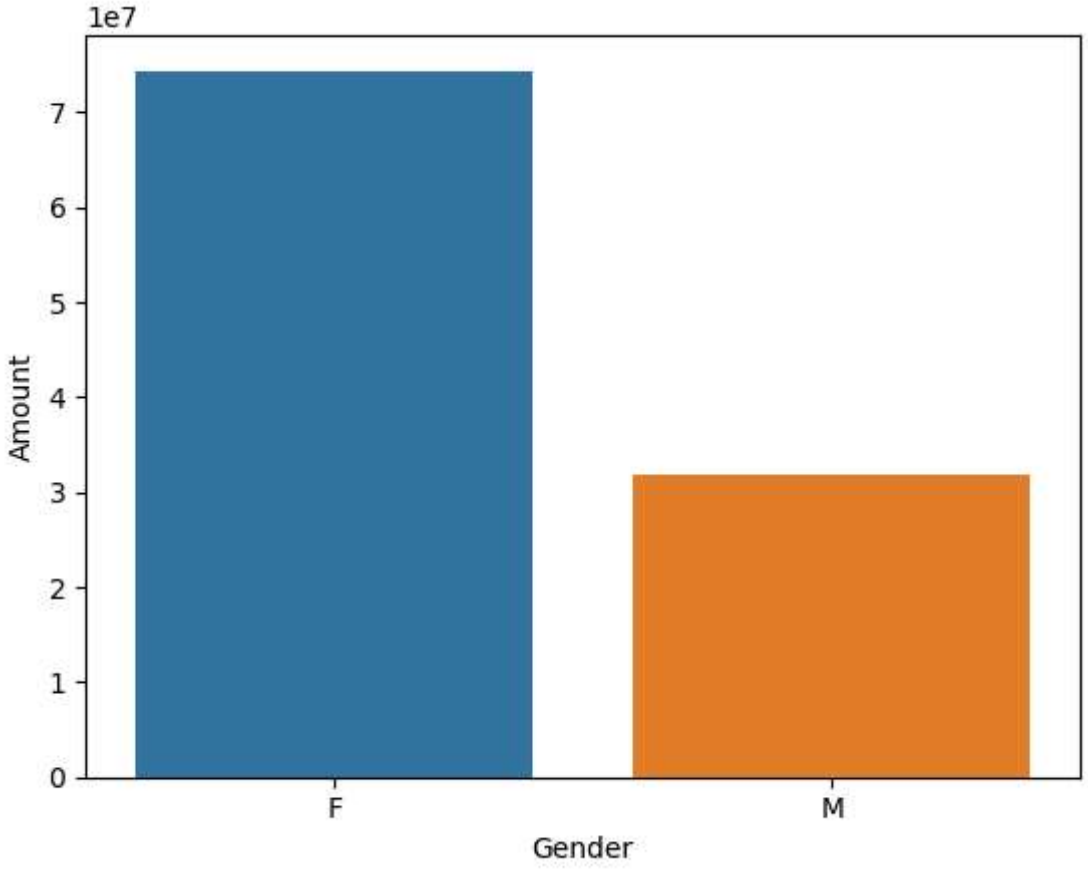
	Age	Amount	Orders
count	11239.000000	11239.000000	11239.000000
mean	35.410357	9453.610553	2.489634
std	12.753866	5222.355168	1.114967
min	12.000000	188.000000	1.000000
25%	27.000000	5443.000000	2.000000
50%	33.000000	8109.000000	2.000000
75%	43.000000	12675.000000	3.000000
max	92.000000	23952.000000	4.000000

In [21]:
```
#Exploratory Data Analysis
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

for bars in ax.containers:
    ax.bar_label(bars)
```

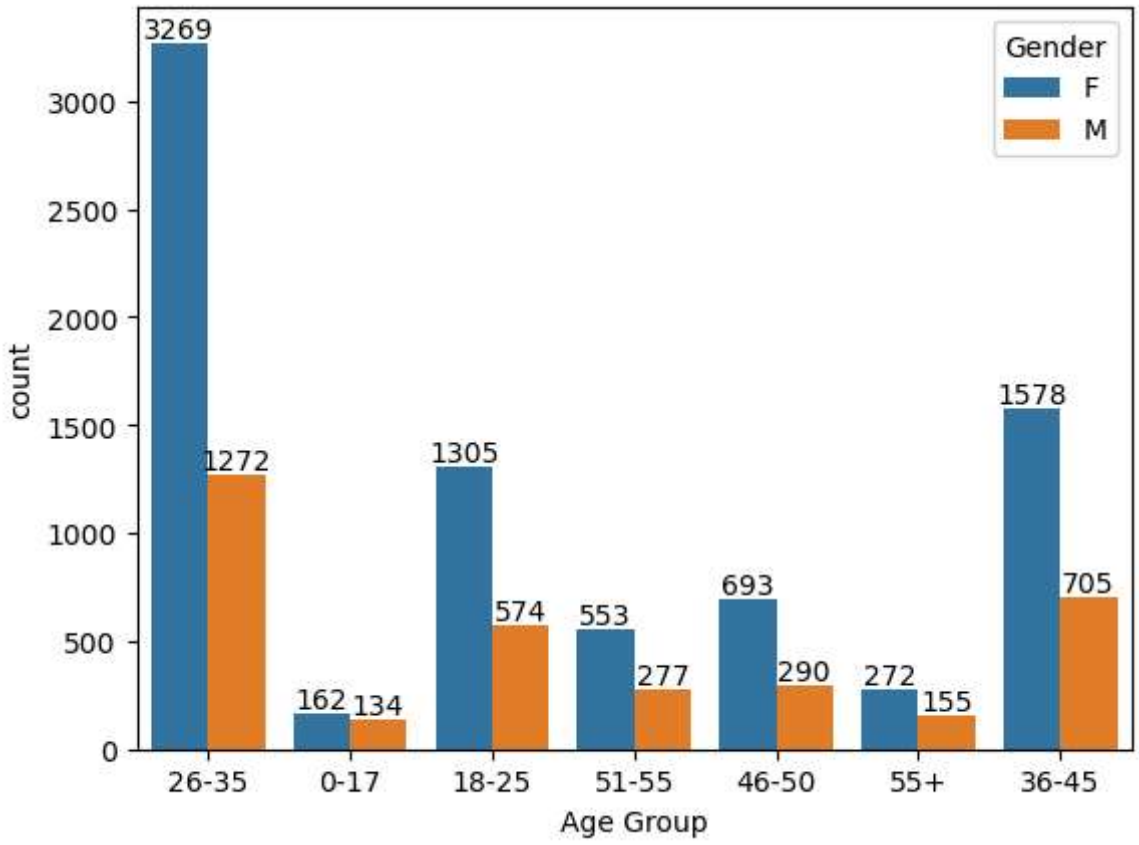


In [22]:
```
# plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amo

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

Out[22]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```

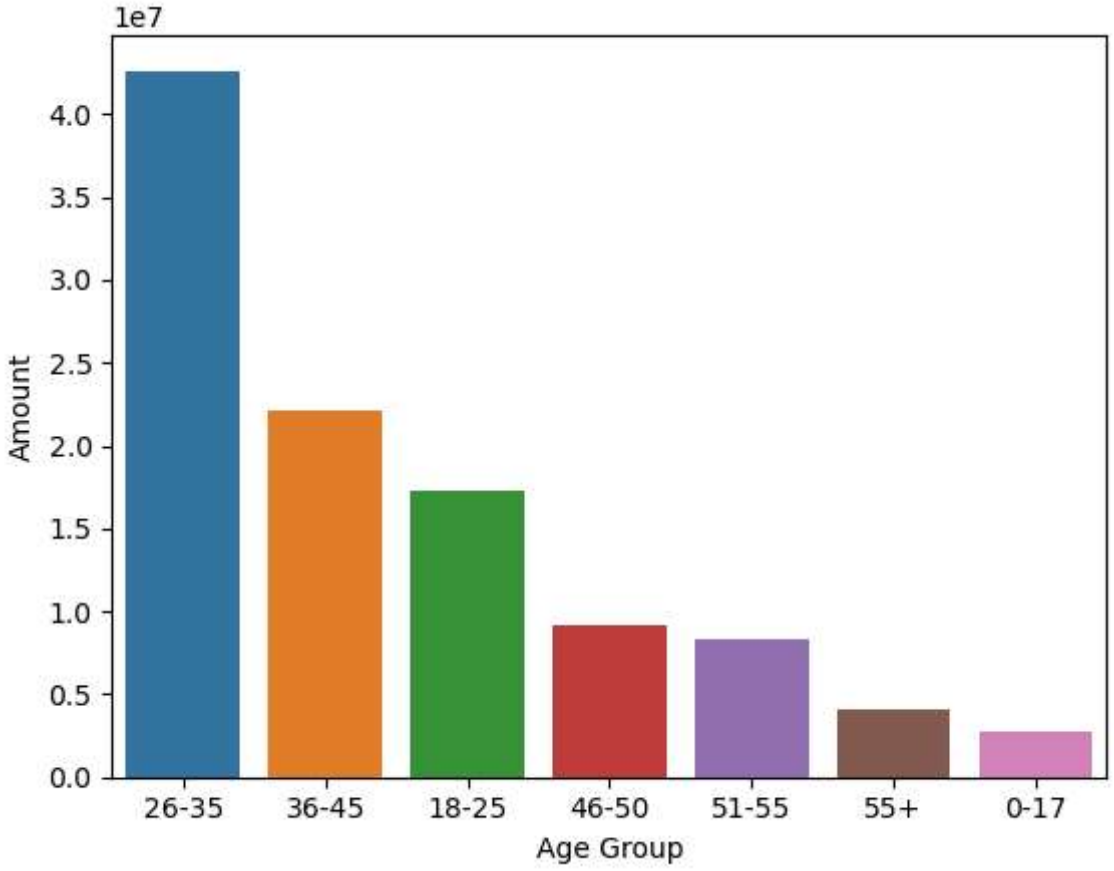


In [23]:
```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

for bars in ax.containers:
    ax.bar_label(bars)
```



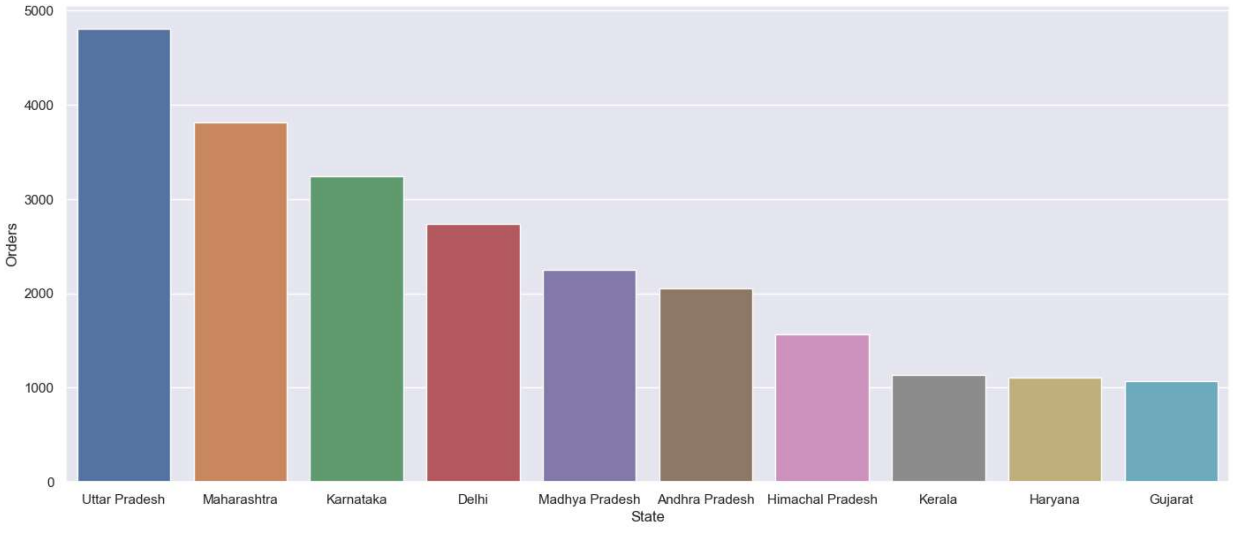
In [24]:
```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by='

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

Out[24]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



In [27]:
```
# total number of orders from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Or

sns.set(rc={'figure.figsize':(17,7)})
sns.barplot(data = sales_state, x = 'State',y= 'Orders')
```

Out[27]:
```
<Axes: xlabel='State', ylabel='Orders'>
```

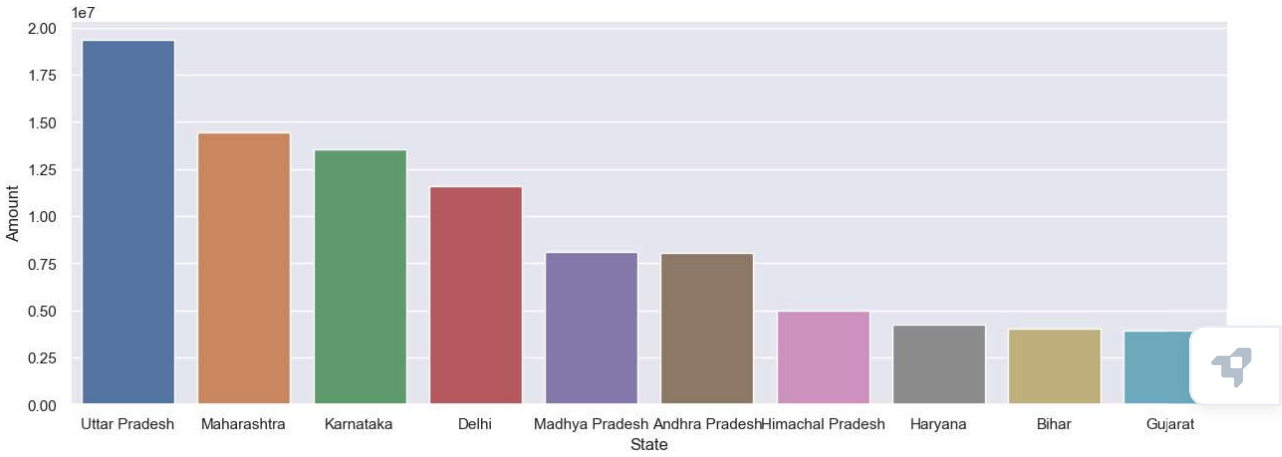


In [31]:
```
# total amount/sales from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Am
```

```
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Amount')
```

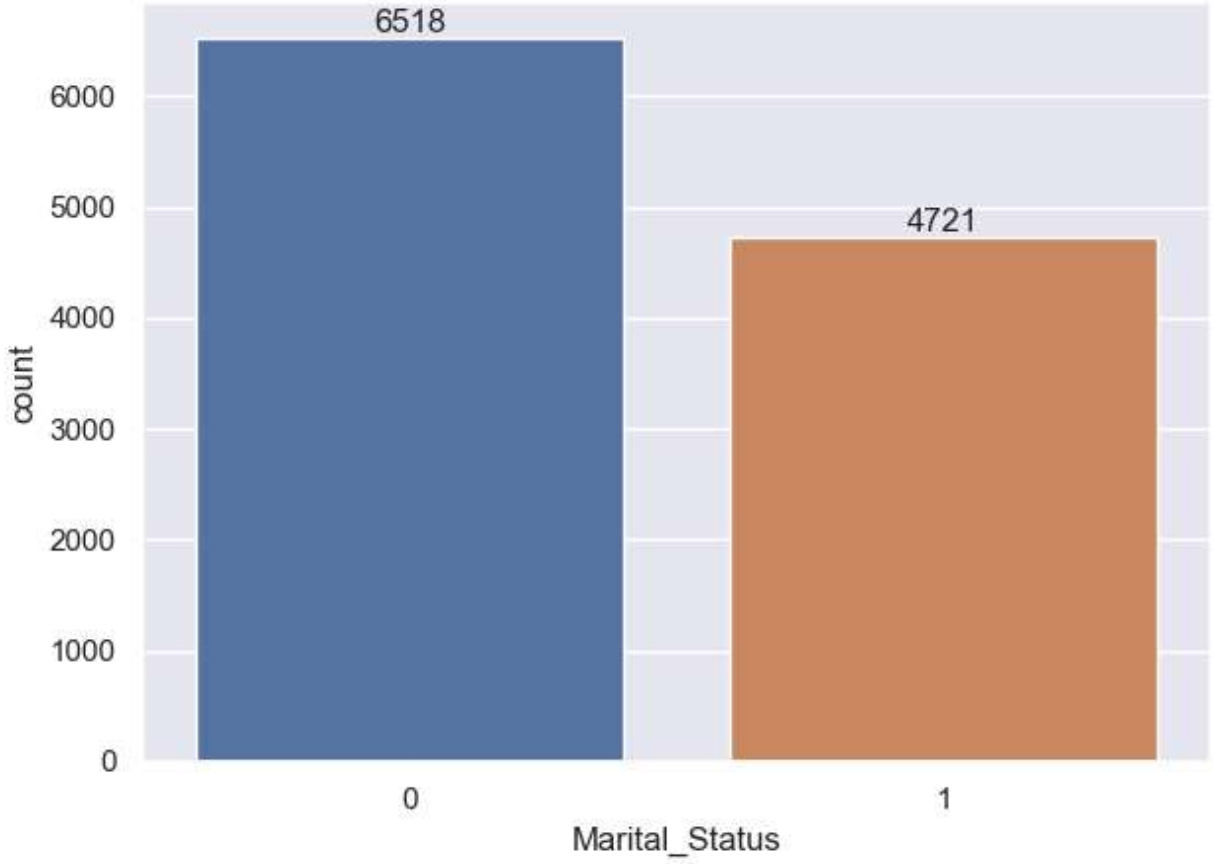
Out[31]:

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



In [29]:

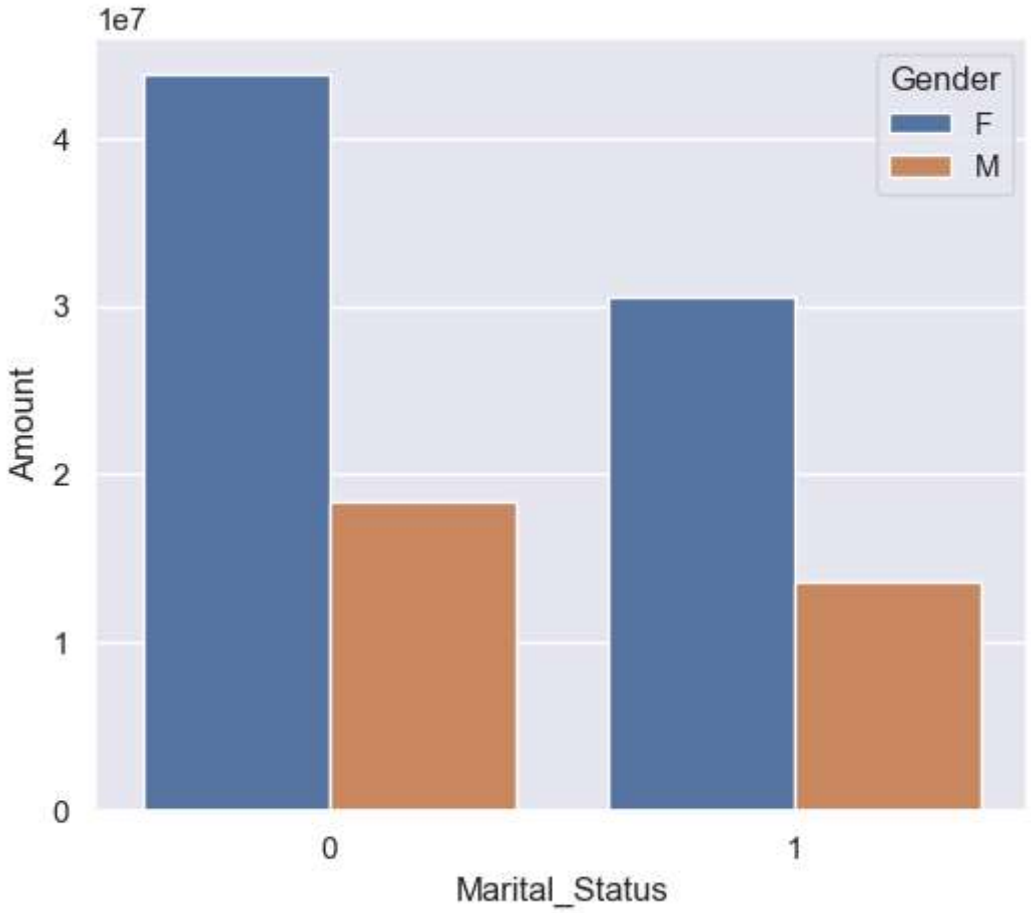
```
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize':(10,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



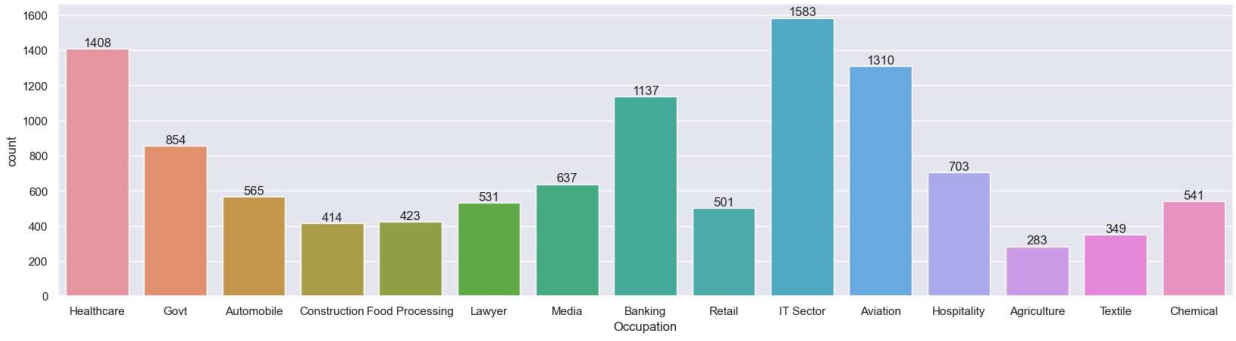
In [30]:

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

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender')
```

Out[30]: <Axes: xlabel='Marital_Status', ylabel='Amount'>

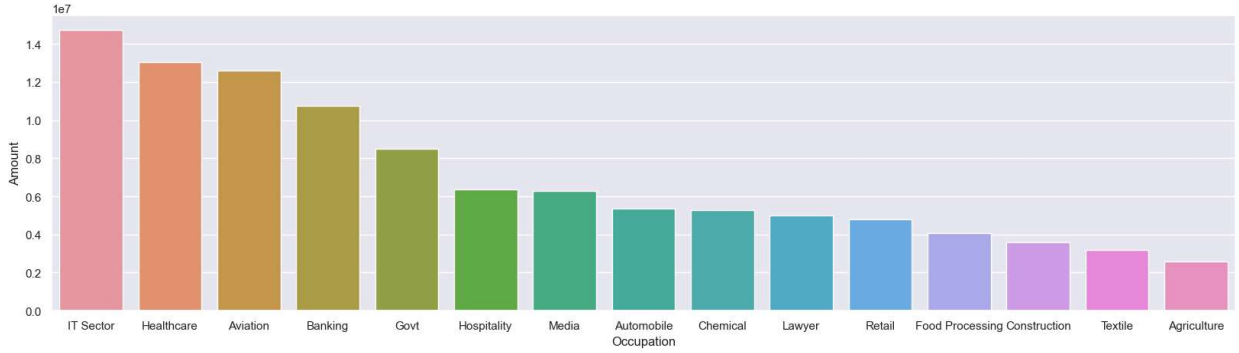


In [32]:
```
#Occupation
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

for bars in ax.containers:
    ax.bar_label(bars)
```

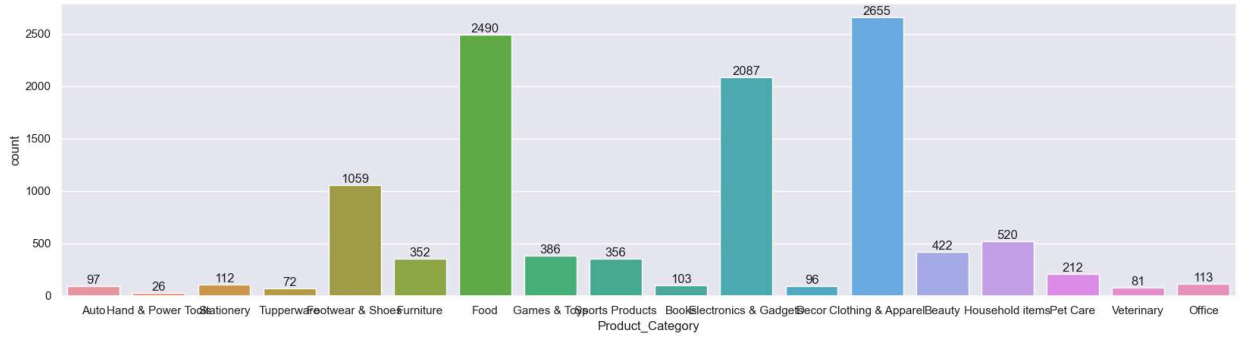


In [34]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(b

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

Out[34]: <Axes: xlabel='Occupation', ylabel='Amount'>

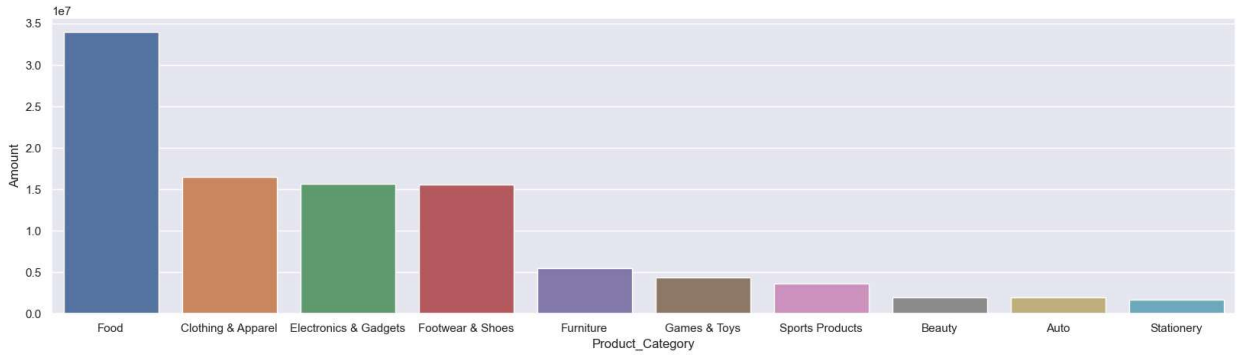


In [35]:
```
#Product Category
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

for bars in ax.containers:
    ax.bar_label(bars)
```



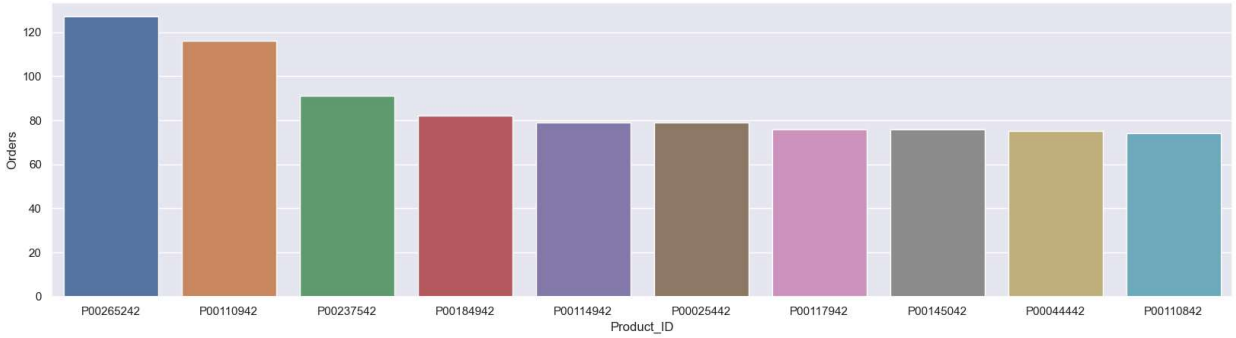
In [36]:
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_va

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount')
```

Out[36]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



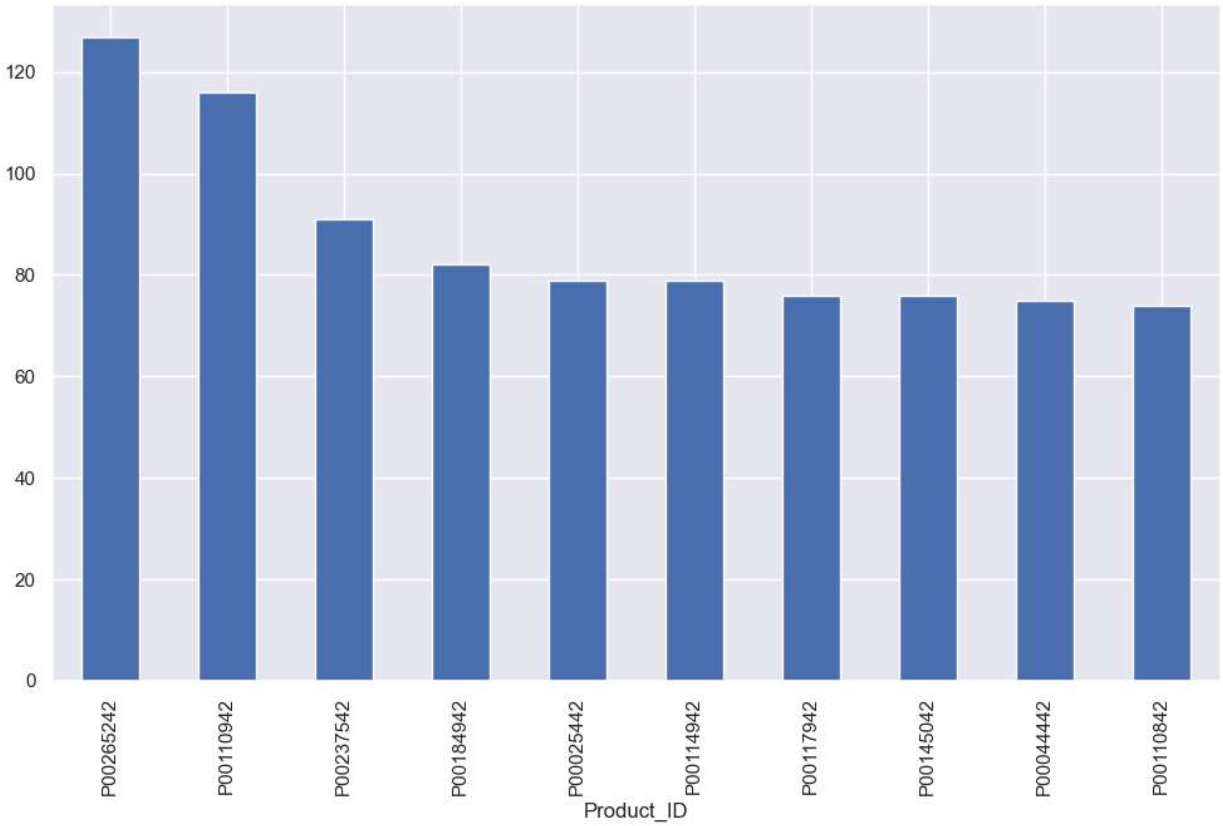
In [37]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(b

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')
```

Out[37]:
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



In [39]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False)
```

Out[39]:
```
<Axes: xlabel='Product_ID'>
```



In [ ]: