In [1]:

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

In [2]:

```
df=pd.read_csv("data files/Gudi_padwa_sales")
df
```

Out[2]:

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

11251 rows × 15 columns

In [3]:

```
df.shape
```

Out[3]:

```
(11251, 15)
```

In [4]:
```
df.head(5)
```

Out[4]:

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

In [5]:
```
df.tail(5)
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	O
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	

In [6]:

```
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 [9]:

```
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [10]:

```
pd.isnull(df).sum()
```

Out[10]:

```
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 [11]:

```
df.shape
```

Out[11]:

```
(11251, 13)
```

In [12]:

```
df.dropna(inplace=True)
```

In [13]:

```
df.shape
```

Out[13]:

```
(11239, 13)
```

In [19]:

```
data_test=[['madhav',11],['keshav',],['lalita',16]]
df_test=pd.DataFrame(data_test,columns=['Name','Age'])
df_test
```

Out[19]:

	Name	Age
0	madhav	11.0
1	keshav	NaN
2	lalita	16.0

In [20]:

```
df_test.dropna()
```

Out[20]:

	Name	Age
0	madhav	11.0
2	lalita	16.0

In [22]:

```
df_test
```

Out[22]:

	Name	Age
0	madhav	11.0
1	keshav	NaN
2	lalita	16.0

In [23]:

```
# change data types.
df['Amount']=df['Amount'].astype('int')
```

In [24]:

```
df['Amount'].dtypes
```

Out[24]:

```
dtype('int32')
```

In [25]:

```
df.columns
```

Out[25]:

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

```
In [27]: df.rename(columns={'Marital_Status':'Married'})
```

Out[27]:

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

11239 rows × 13 columns

In [28]:

```
df.describe()
```

Out[28]:

	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	2.000000	5443.000000
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

In [29]:

```
df[['Age','Orders','Amount']].describe()
```

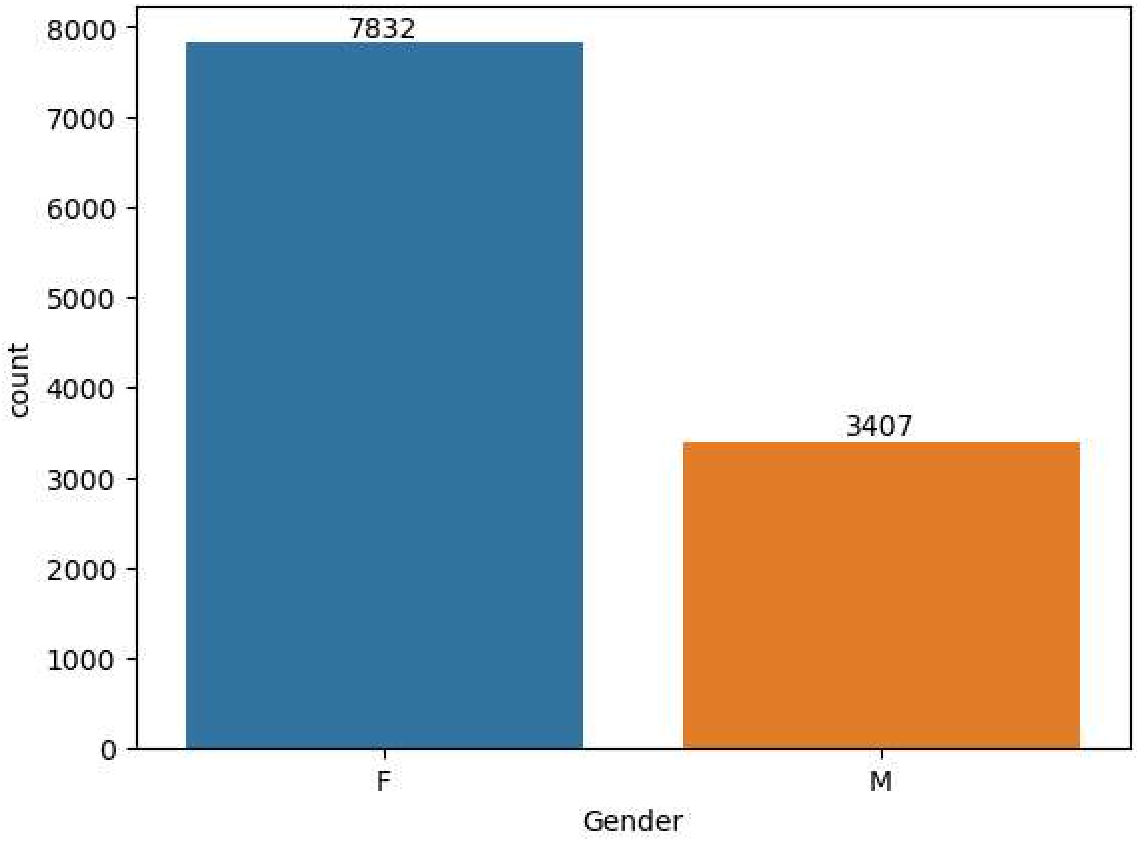
Out[29]:

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

#Exploratory_Data_Analysis

In [32]:

```
#Gender
ax=sns.countplot(x='Gender',data=df)
for bars  in ax.containers:
    ax.bar_label(bars)
```

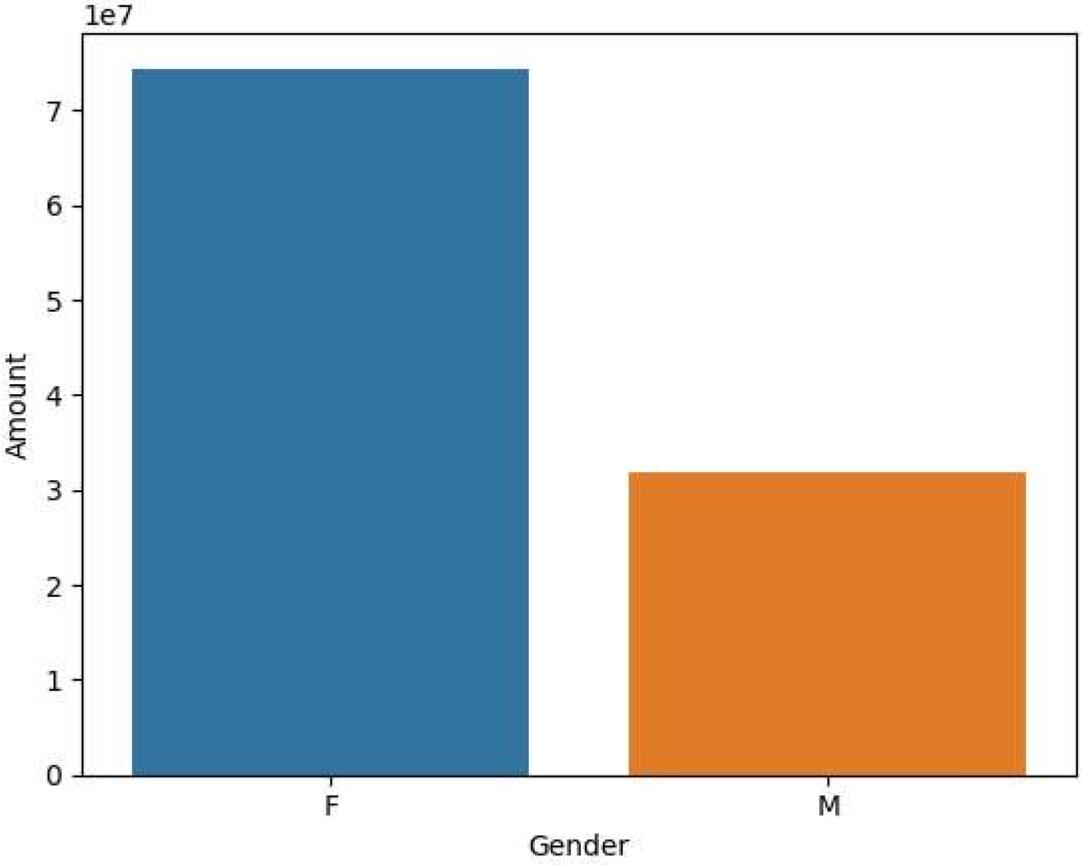


In [33]:

```
sales_gen=df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Gender',y='Amount',data=sales_gen)
```

Out[33]:

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

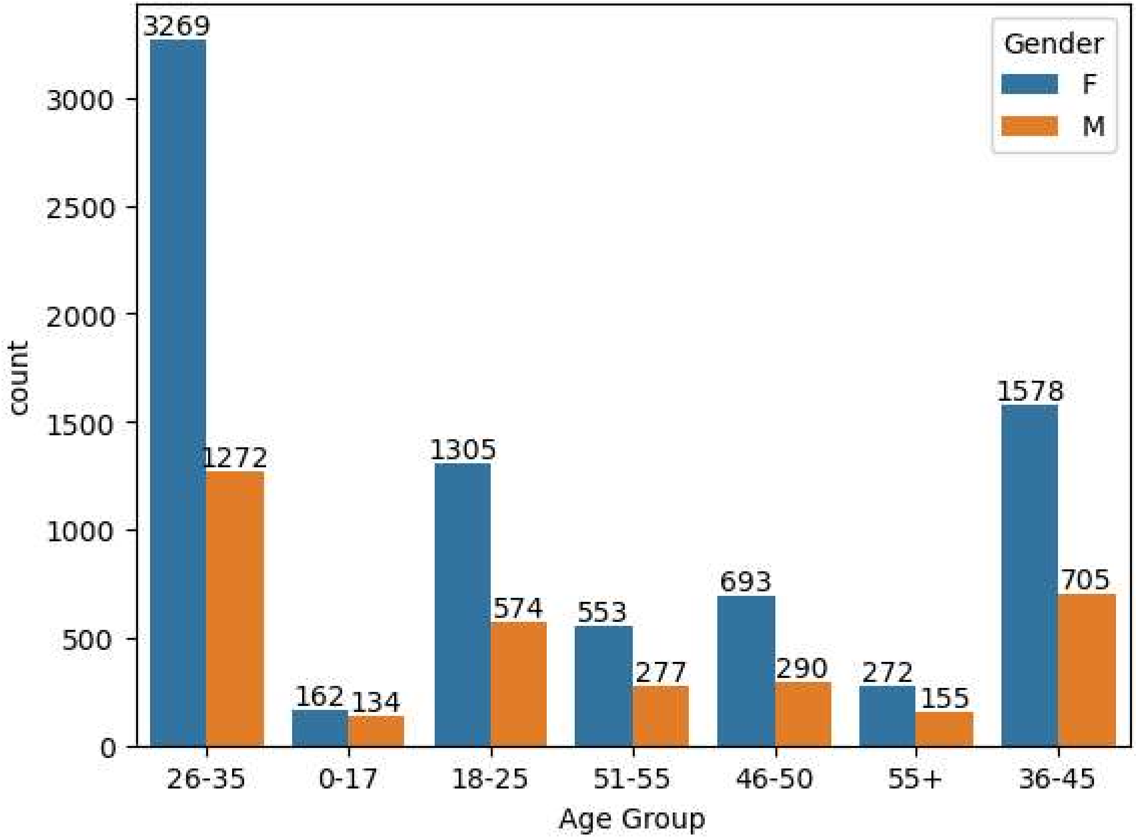


In [34]:

```
# as we know females are most of the buyers and even purchaing power of females are gerater than male.
```

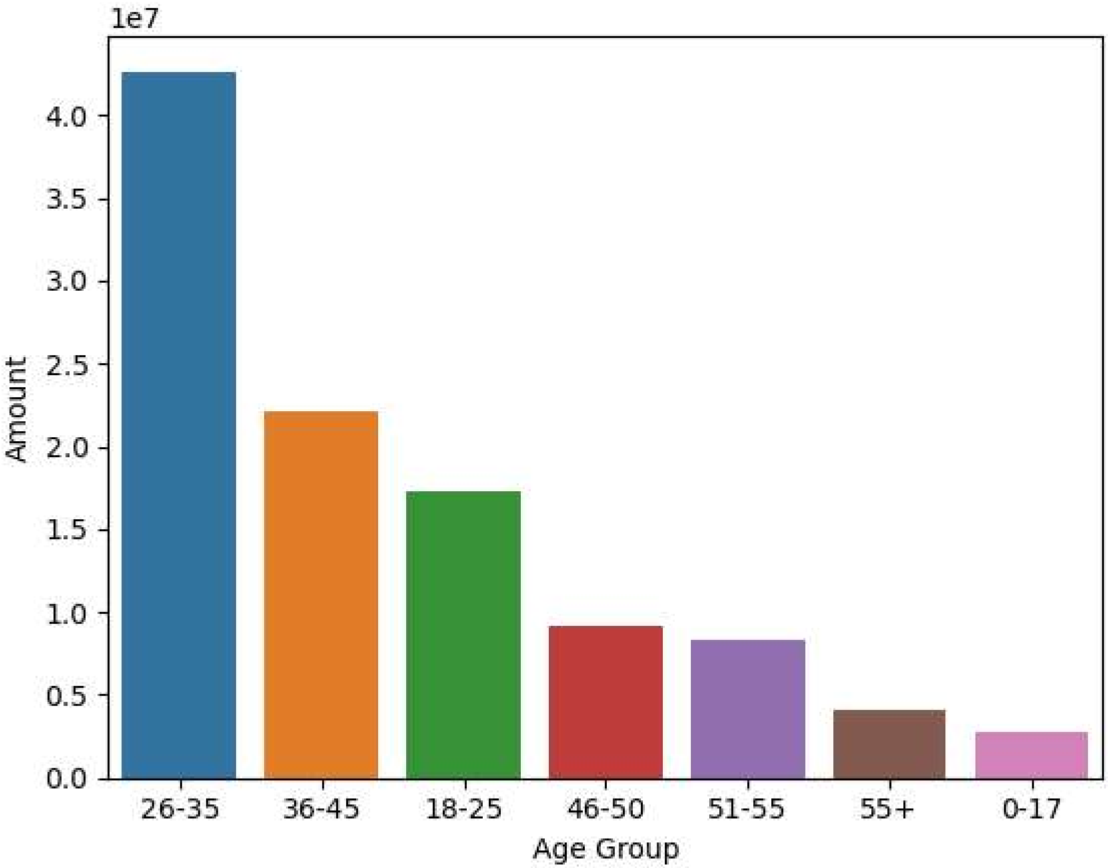
In [36]:

```
#Age
ax=sns.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [37]: sales_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
         sns.barplot(x='Age Group',y='Amount',data=sales_age)
```

```
Out[37]: <Axes: xlabel='Age Group', ylabel='Amount'>
```

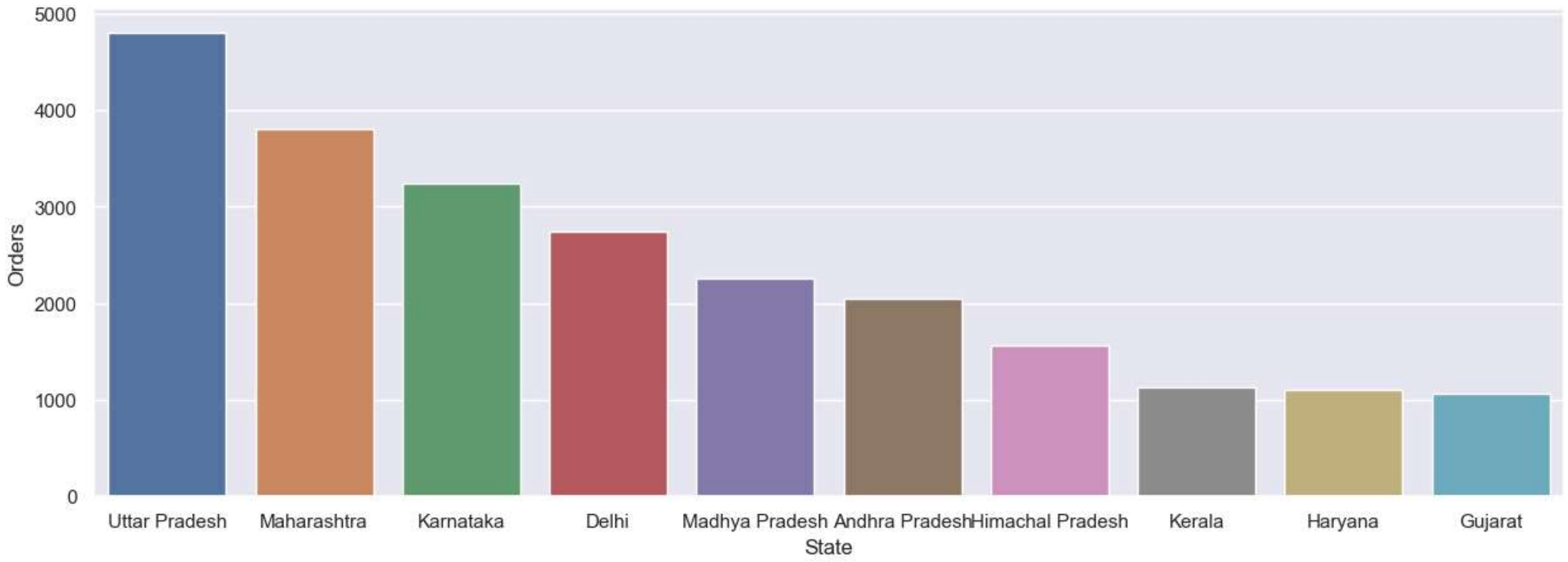


```
In [38]: #state
         df.columns
```

```
Out[38]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
                'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
                'Orders', 'Amount'],
               dtype='object')
```

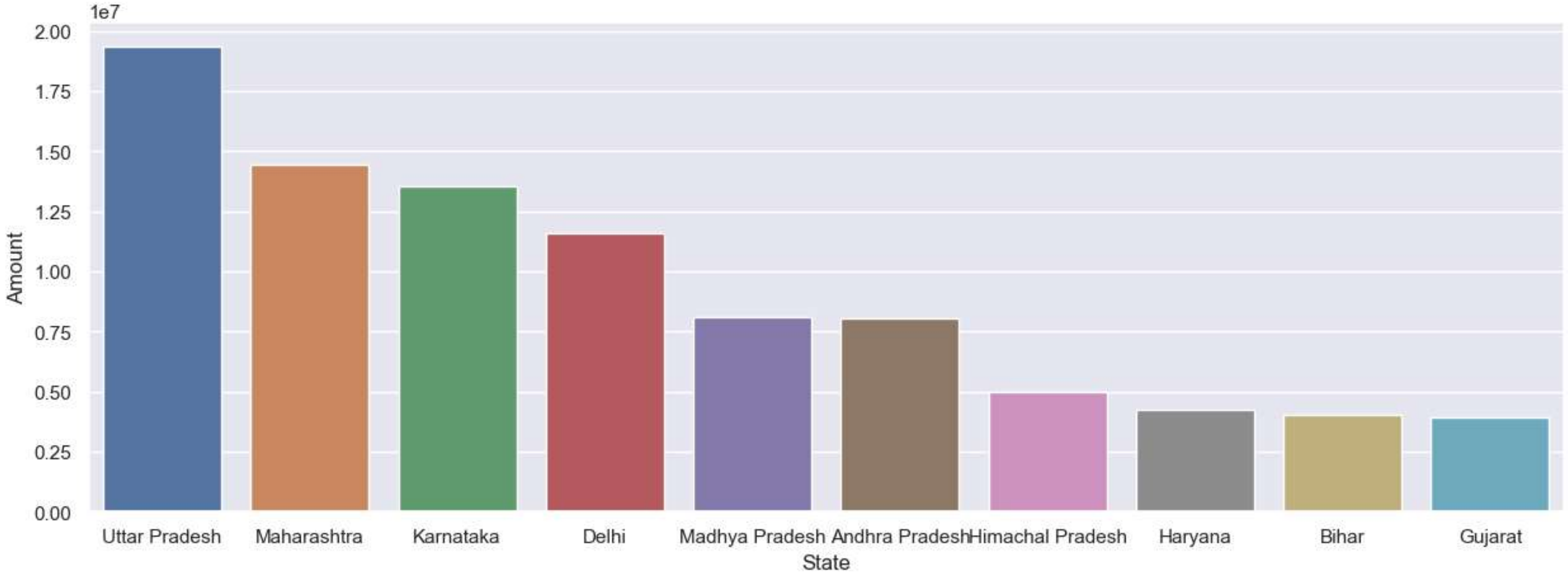
In [43]:

```
#total numbers of oreders from top 10 states
sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).hea
sns.barplot(data=sales_state,x='State',y='Orders')
sns.set(rc={'figure.figsize':(15,5)})
```



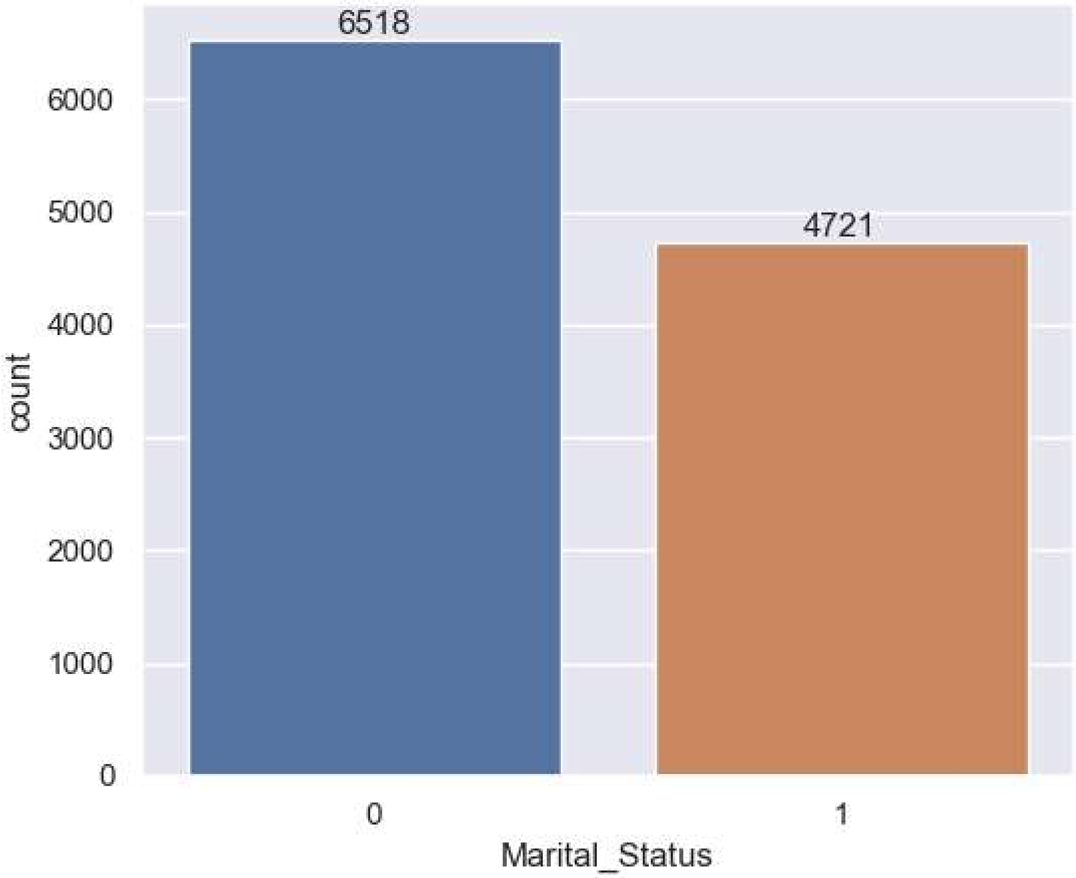
In [45]:
```
# total amount  from top 10 states
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).hea
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state,x='State',y='Amount')
```

Out[45]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



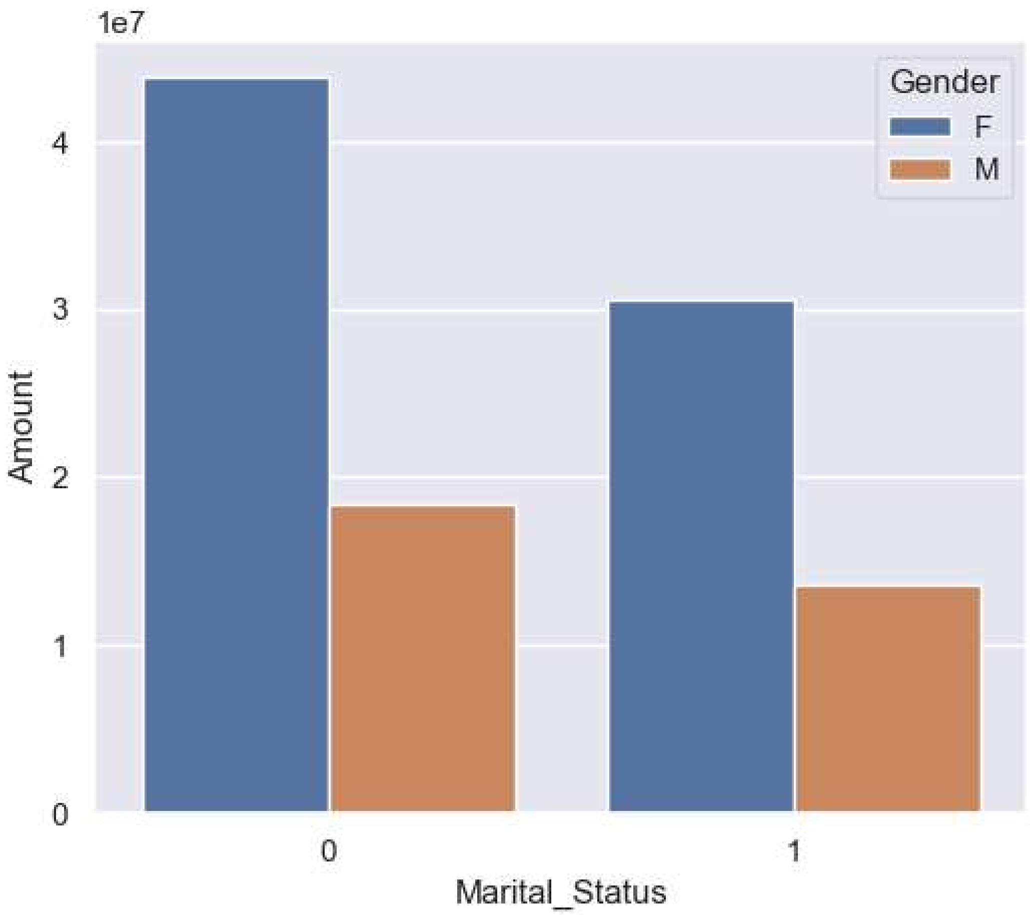
In [53]:

```
# martial status
ax=sns.countplot(data=df,x='Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [51]:
```
sales_state=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',as
sns.set(rc={'figure.figsize':[6,5]})
sns.barplot(data=sales_state,x="Marital_Status",y="Amount",hue="Gender")
```

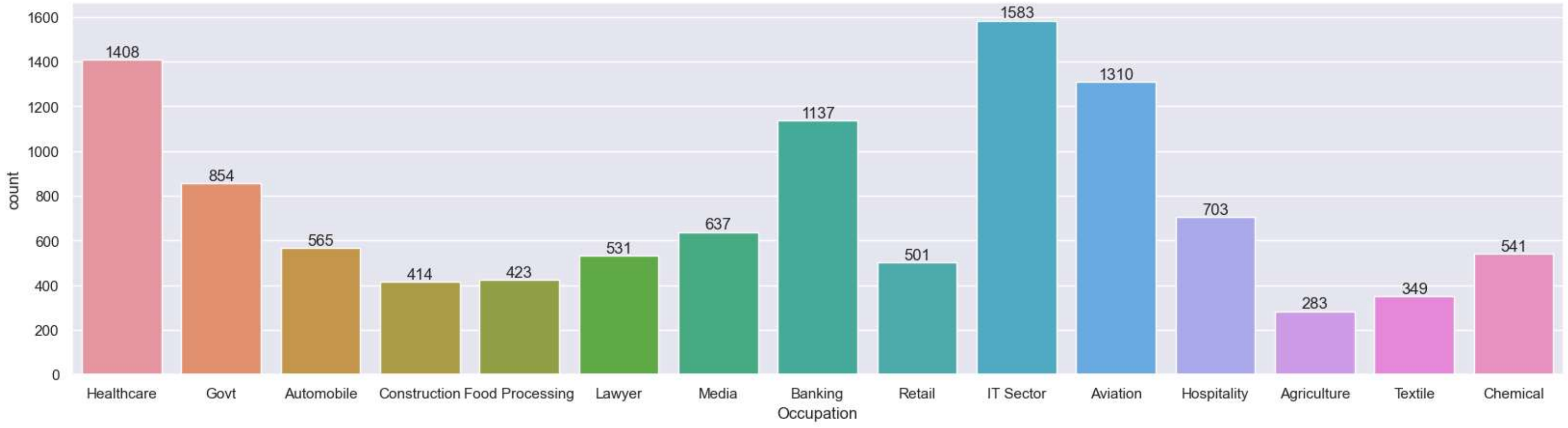
Out[51]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



In [54]:
```
#occupation
```

In [55]:

```
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data=df,x='Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```

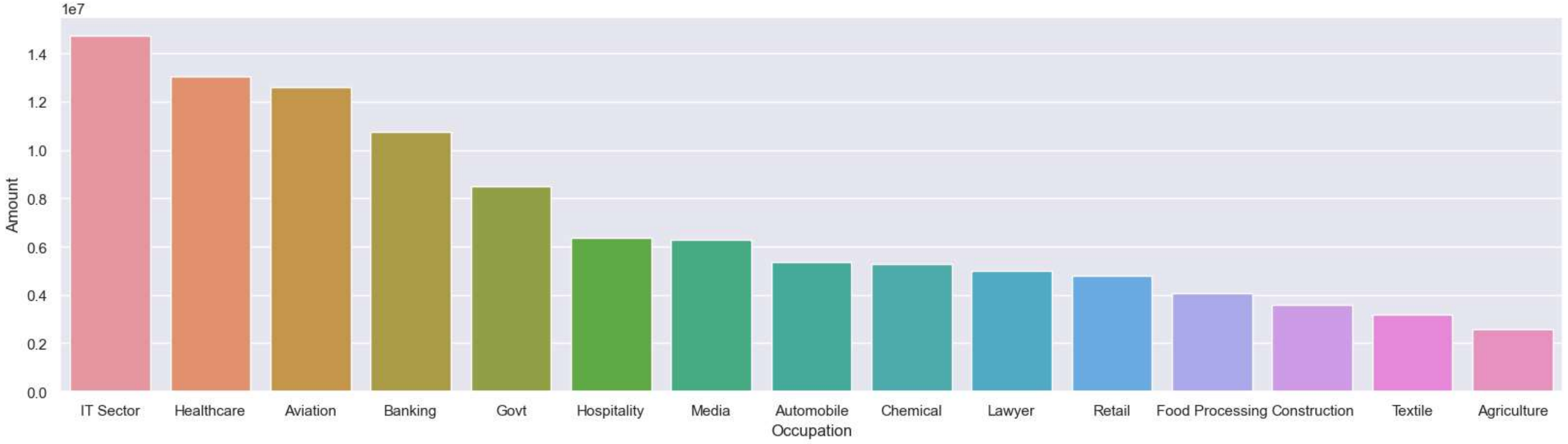


In [57]:

```
sales_state=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False
sns.set(rc={'figure.figsize':[20,5]})
sns.barplot(data=sales_state,x="Occupation",y="Amount")
```

Out[57]:

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

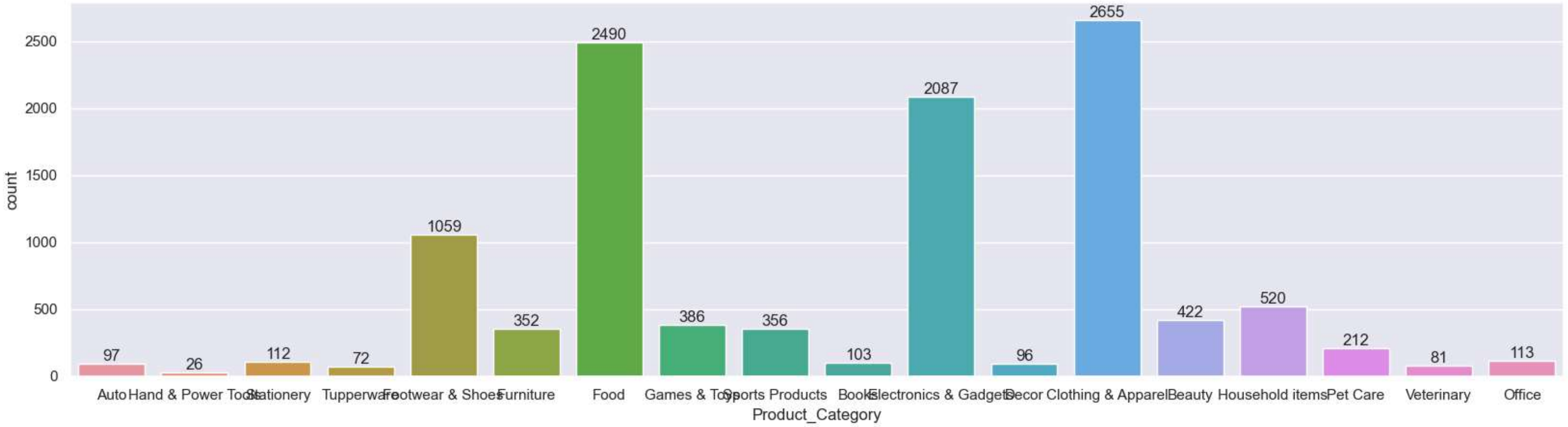


```
In [58]: df.columns
```

```
Out[58]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
               'Orders', 'Amount'],
              dtype='object')
```

```
In [59]: # product category
```

```
In [62]: sns.set(rc={'figure.figsize':(20,5)})
         ax=sns.countplot(data=df,x='Product_Category')
         for bars in ax.containers:
             ax.bar_label(bars)
```

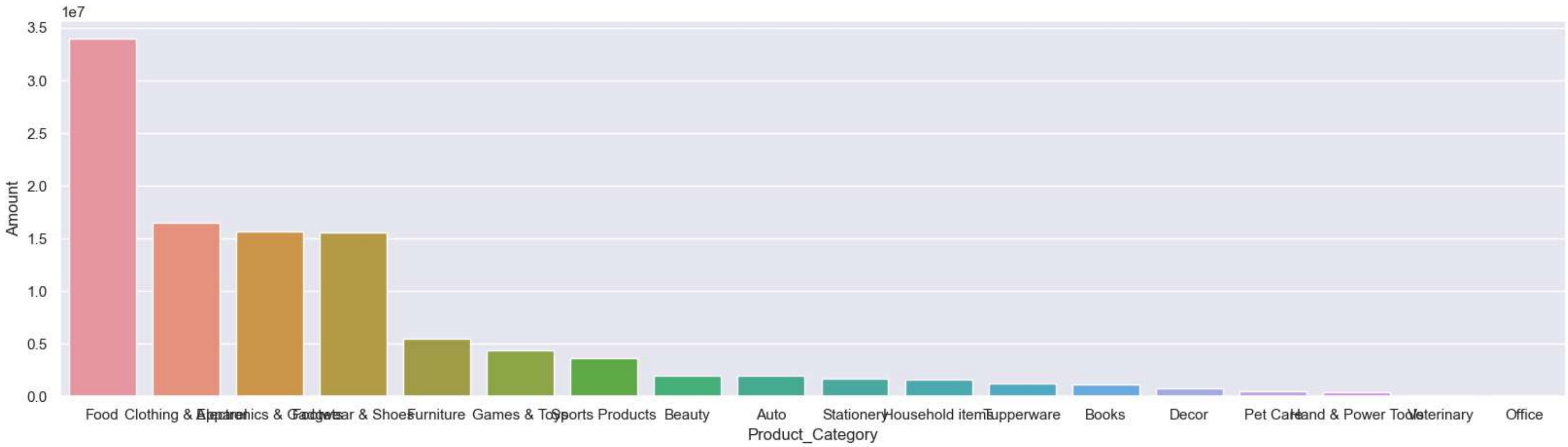


In [66]:

```
sales_state=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state,x='Product_Category',y='Amount')
```

Out[66]:

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

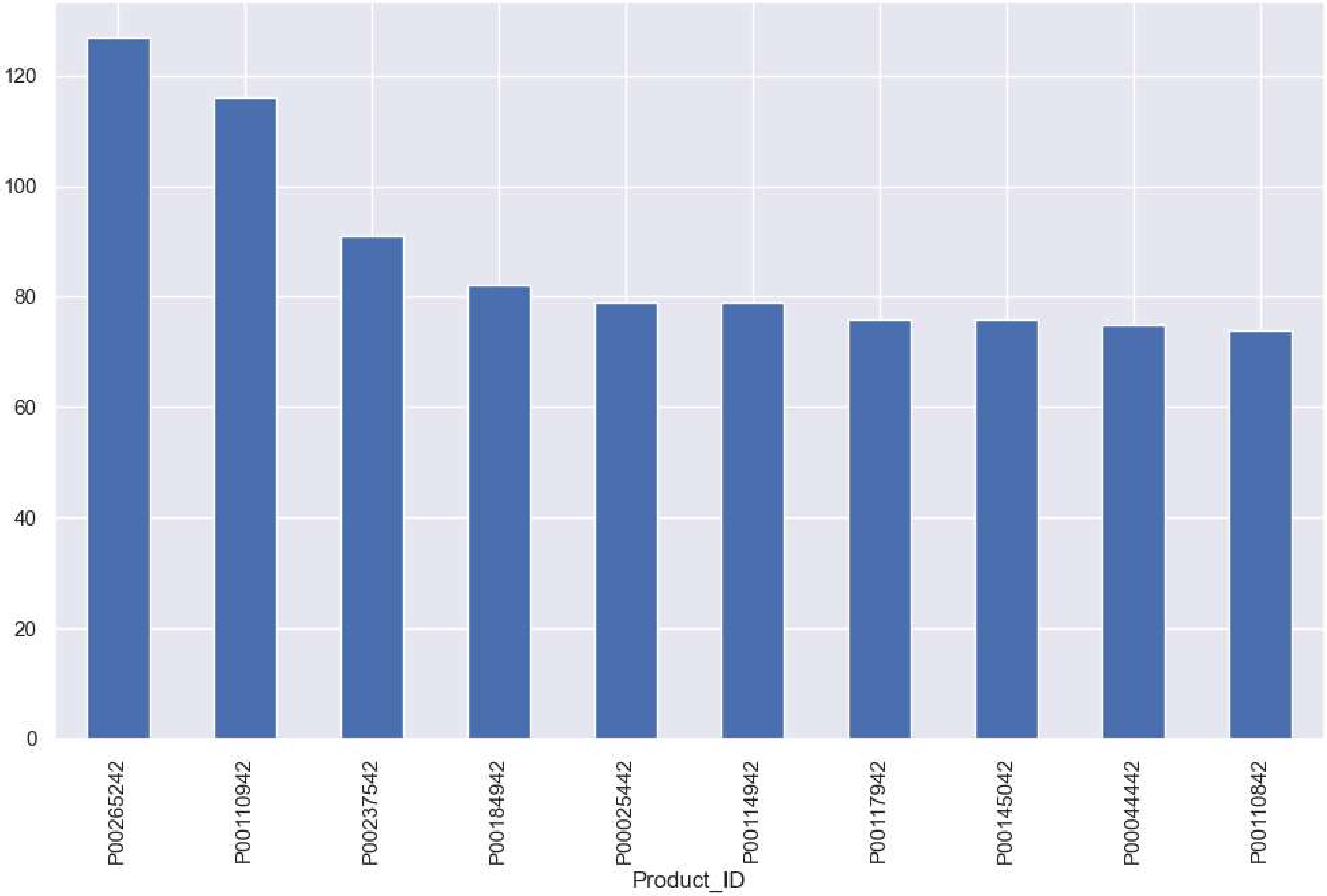


In [67]:

```
# top 10 most sold products
```

```
In [71]: fig1,ax1=plt.subplots(figsize=(12,7))
         df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')

Out[71]: <Axes: xlabel='Product_ID'>
```



```
In [ ]:
```