In [ ]:
```
#Importing all important libraies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

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

In [5]:
```
df.shape
```

Out[5]: (11251, 15)

In [8]:
```
df.head(20)
```

Out[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zo
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	West
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	South
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Cen
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	South
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	West
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	North
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Cen
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	West
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Cen
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	South
10	1003829	Harshita	P00200842	M	26-35	34	0	Delhi	Cen
11	1000214	Kargatis	P00119142	F	18-25	20	0	Andhra Pradesh	South
12	1004035	Elijah	P00080342	F	18-25	20	1	Andhra Pradesh	South
13	1001680	Vasudev	P00324942	M	26-35	26	1	Andhra Pradesh	South
14	1003858	Cano	P00293742	M	46-50	46	1	Madhya Pradesh	Cen
15	1000813	Lauren	P00289942	F	18-25	24	0	Andhra Pradesh	South
16	1005447	Amy	P00275642	F	46-50	48	1	Andhra Pradesh	South
17	1001193	Mick	P00004842	F	26-35	29	0	Andhra Pradesh	South
18	1001883	Praneet	P00029842	M	51-55	54	1	Uttar Pradesh	Cen
19	1001883	Praneet	P00029842	M	51-55	54	1	Uttar Pradesh	Cen

In [10]:
```
df.info(10)
```

```
<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 [11]:
```
#Removing or droping empty columns
df.drop(['Status','unnamed1'],axis=1, inplace=True)
```

In [13]:
```
#Checking the null values
pd.isnull(df).sum()
```

Out[13]:
```
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 [14]:
```
#Droping null columns
df.dropna(inplace=True)
```

In [17]:
```

```

Out[17]:
```
dtype('int32')
```

In [42]:
```
#Changing the datatype of Amount attribute
df['Amount']= df['Amount'].astype('int')
```

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

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

In [18]:
```
df.columns
```

Out[18]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [19]:
```
#Renaming the column Marital_Status with Shaadi
df.rename(columns={'Marital_Status':'Shaadi'})
```

Out[19]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Zone
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
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western
...	...	...	...	...	...	...	...	...	...
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
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western

11239 rows × 13 columns

In [20]:
```
df.describe()
```

Out[20]:

	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 [21]:
```
df[['Age','Orders','Amount']].describe()
```

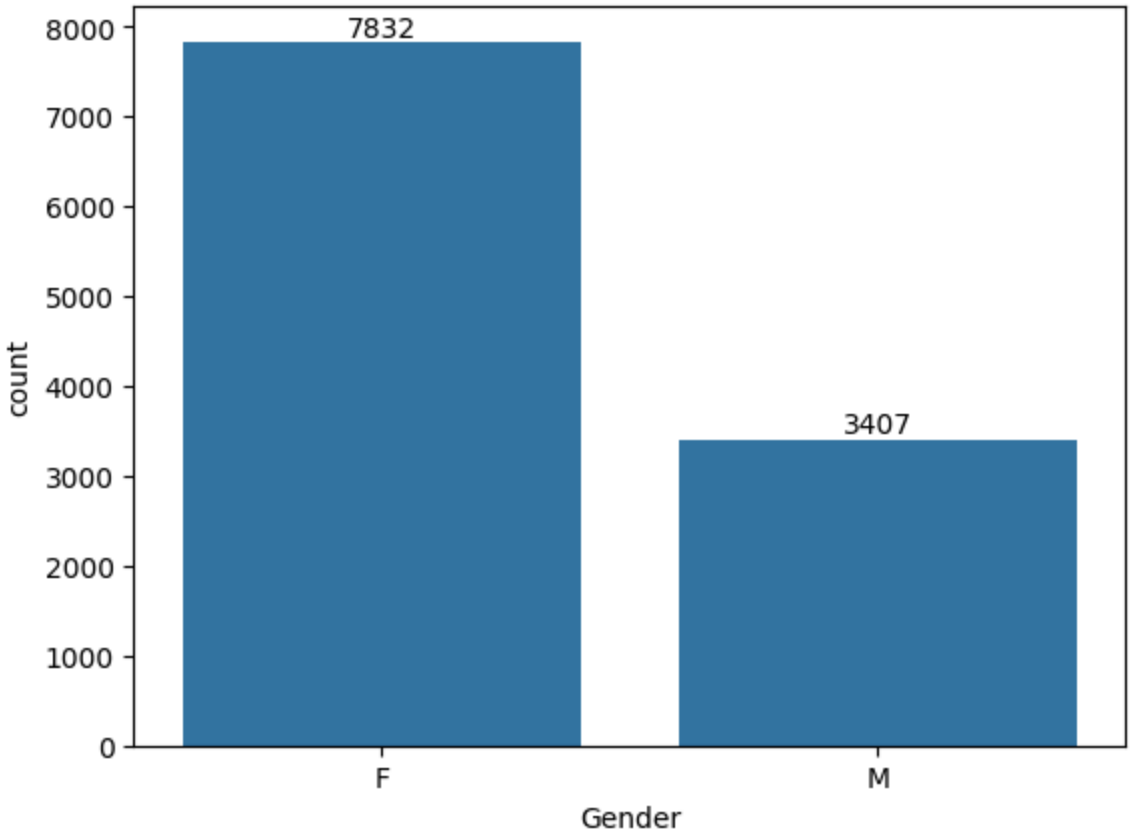
Out[21]:

	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

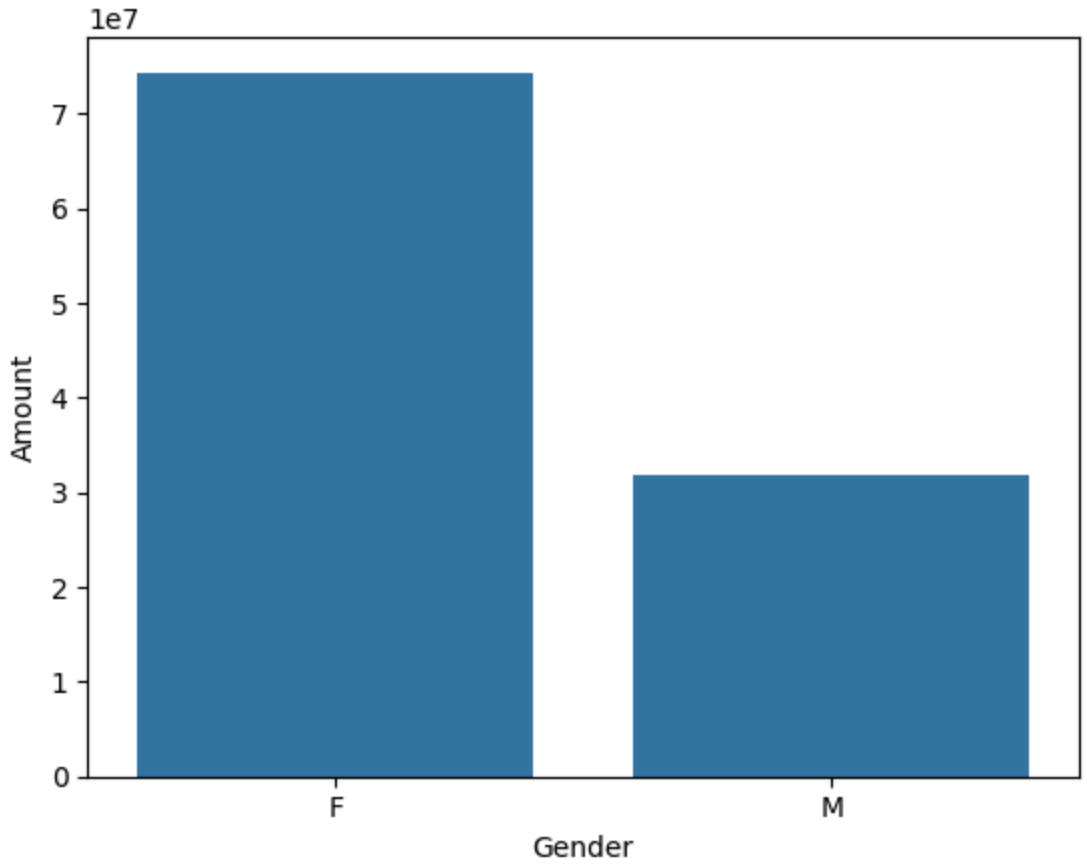
Gender

In [22]:
```
#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 [23]:
```
#Plotting a bar chart for gender vs total amount
sales_gen = df.groupby(['Gender'], as_index= False)['Amount'].sum().sort_values
sns.barplot(x = 'Gender',y='Amount',data = sales_gen)
```

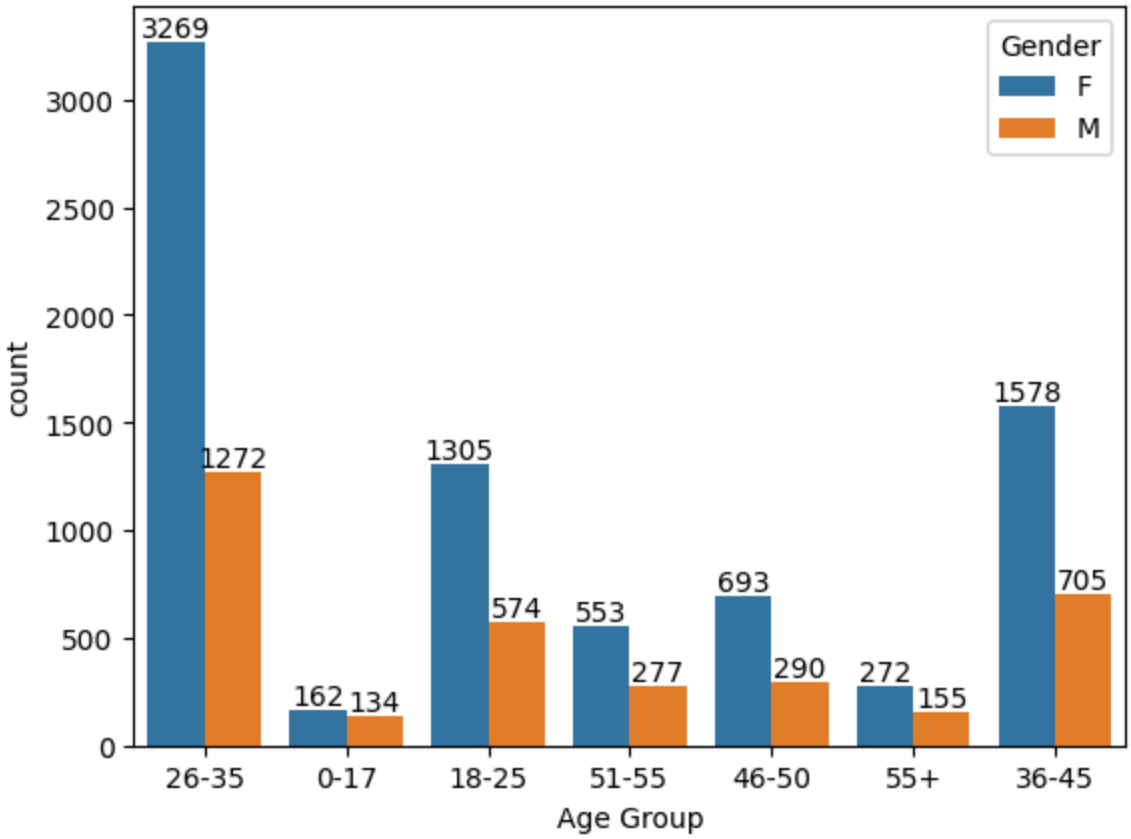
Out[23]: <Axes: xlabel='Gender', ylabel='Amount'>



From above graphs we can see that most of the buyers are females and even the purchasing power of females are gender than men

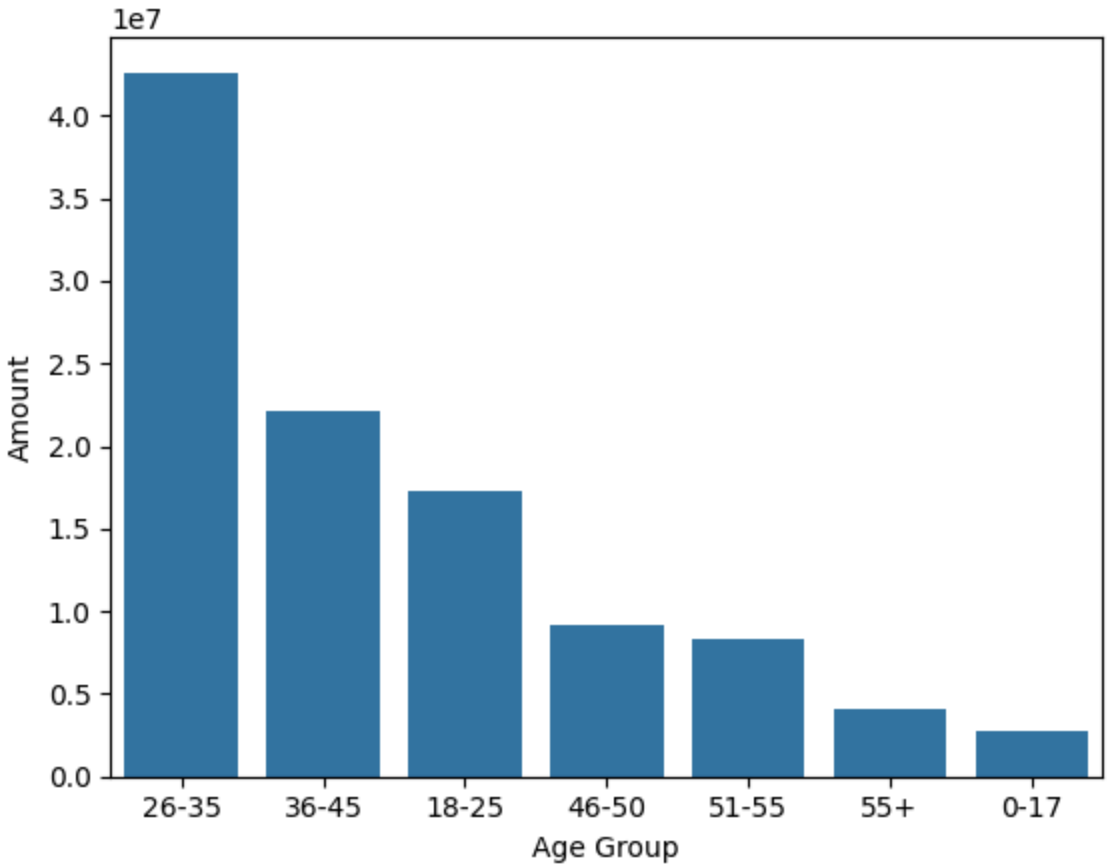
Age

In [25]:
```
ax = sns.countplot(data = df, x = 'Age Group',hue ='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [29]:
```
#Total amount vs Age Group
sales_age =df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_value
sns.barplot(x= 'Age Group',y='Amount', data= sales_age)
```

Out[29]: <Axes: xlabel='Age Group', ylabel='Amount'>

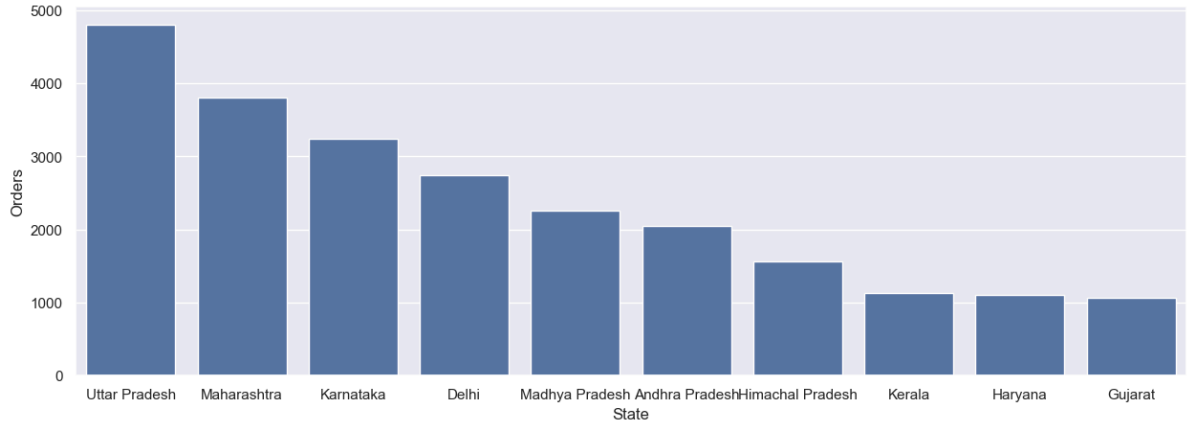


From the above graphs we can see that most of the buyers are of age group between 26-35 yrs female

State

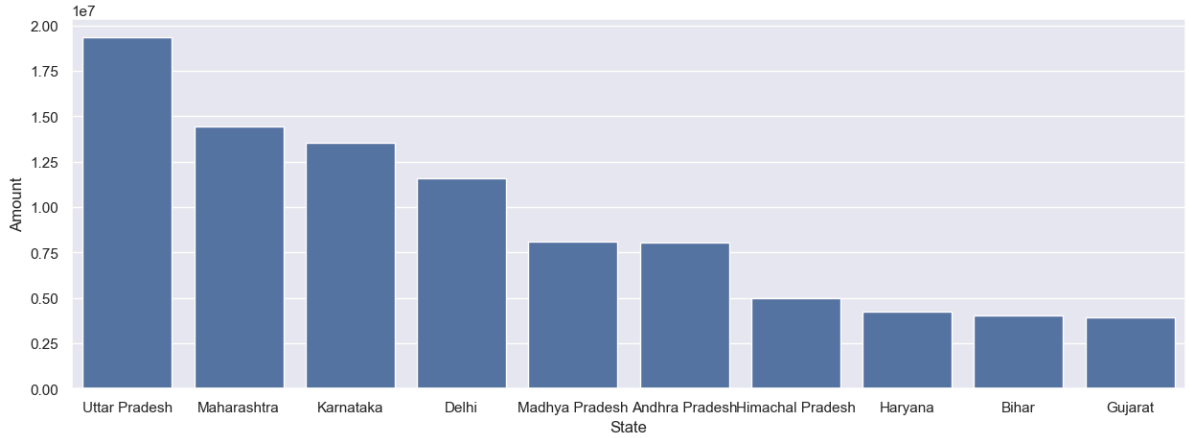
In [31]:
```
#Total number of orders from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state,x='State',y='Orders')
```

Out[31]: <Axes: xlabel='State', ylabel='Orders'>



In [32]:
```
#Total amount/sales from top 10 statesdsdsdfghjasdfqwetuuioopddddddddddddd
sales_state = df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state, x= 'State',y= 'Amount')
```

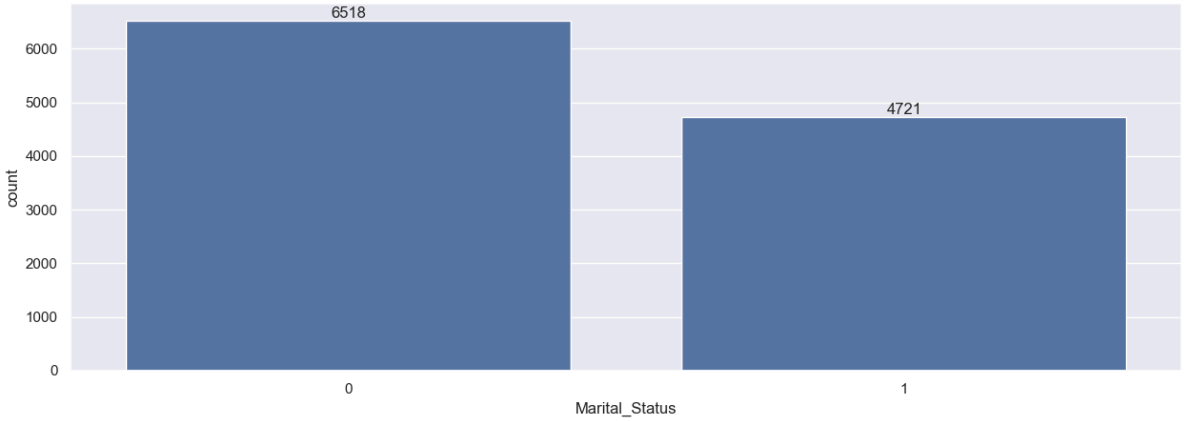
Out[32]: <Axes: xlabel='State', ylabel='Amount'>



From above graphs we can see that most of the orders & total sales/amount are from uttat pradesh, Maharastra and karnataka respectively

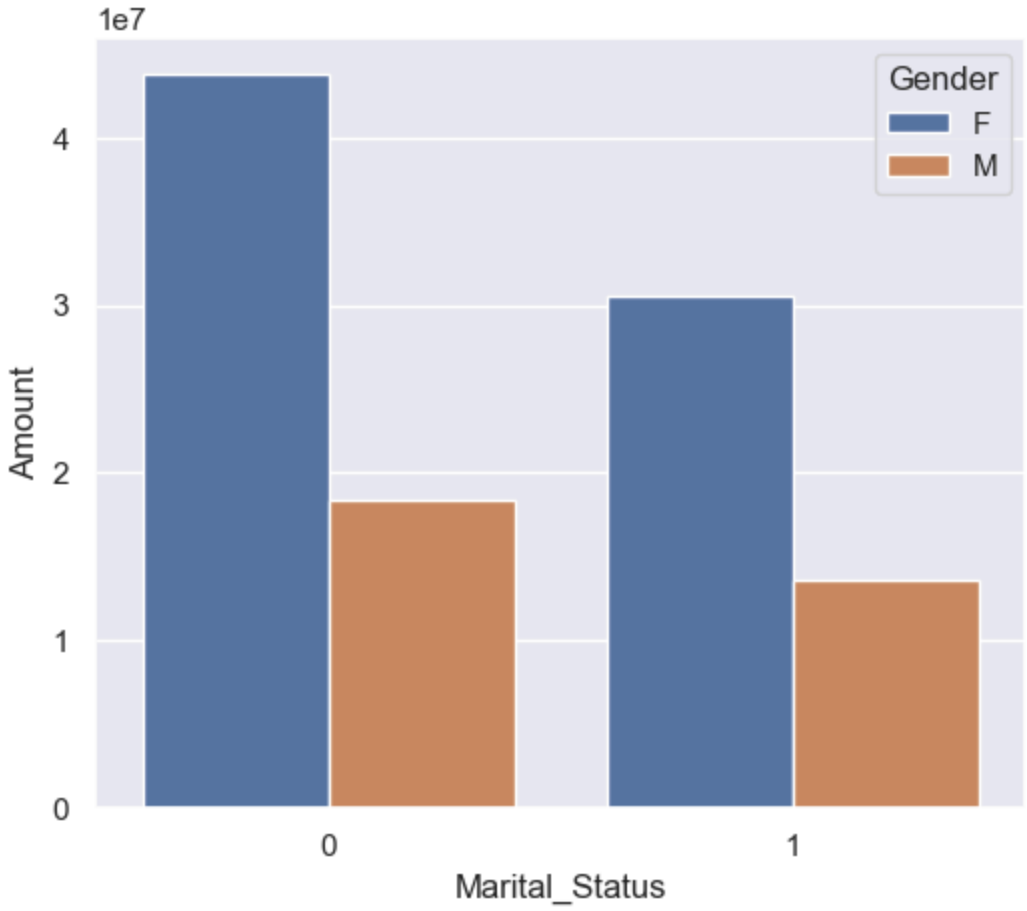
Marital_Status

In [33]:
```
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 [34]:
```
sales_state= df.groupby(['Marital_Status','Gender'], as_index=False)['Amount']
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data= sales_state, x='Marital_Status',y='Amount', hue='Gender')
```

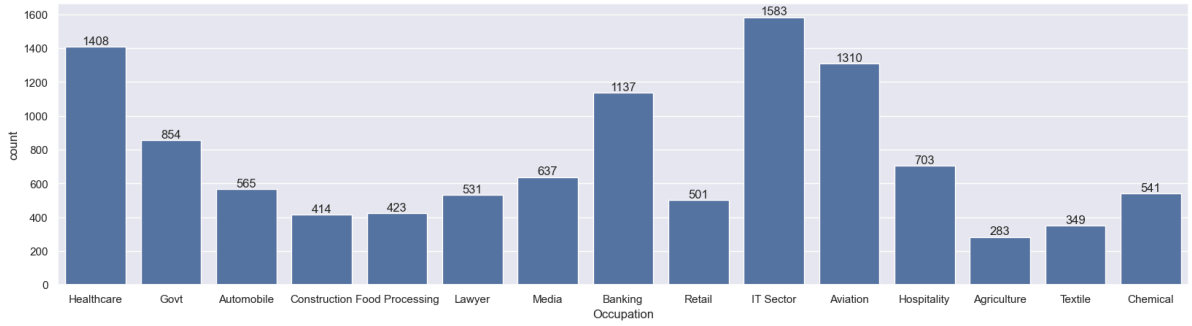
Out[34]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

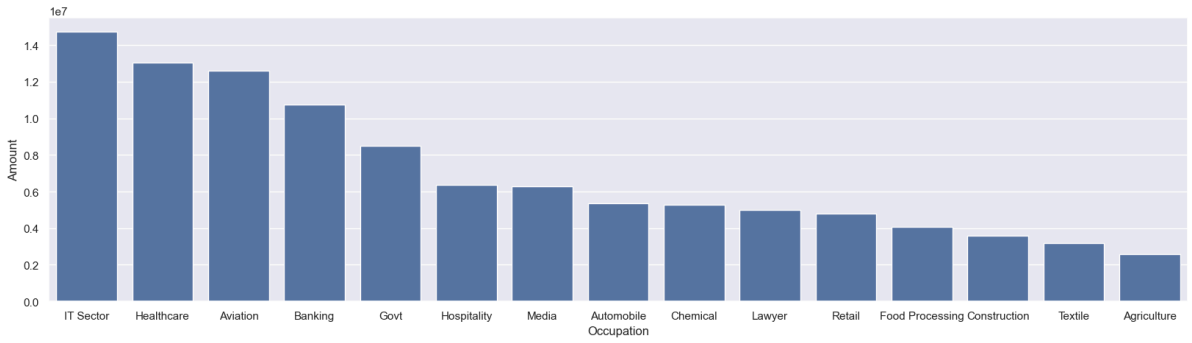
Occupation

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

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



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

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

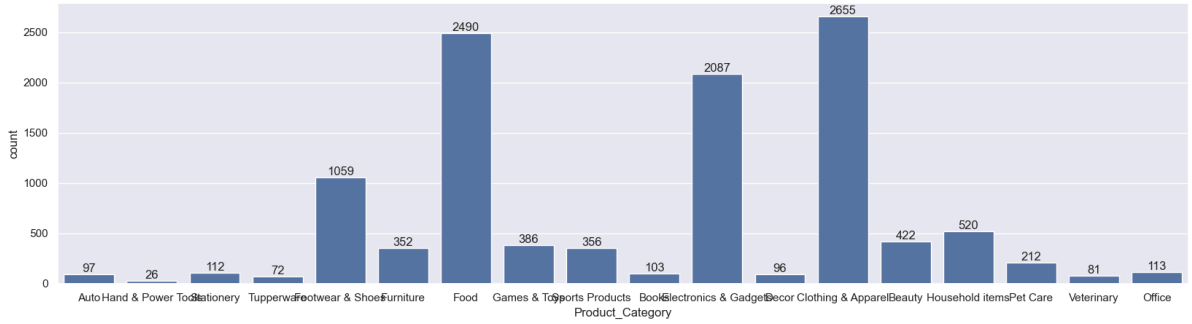
Out[36]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

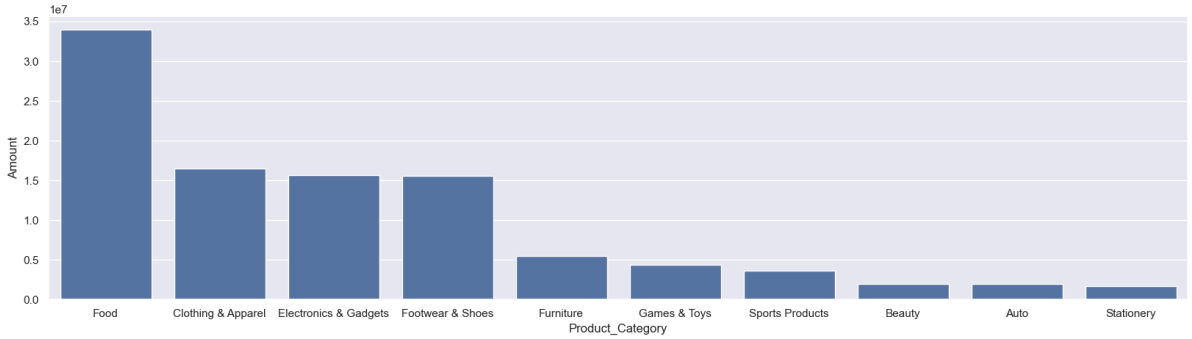
Product Category

In [38]:
```
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 [39]:
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum()

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

Out[39]: <Axes: xlabel='Product_Category', ylabel='Amount'>

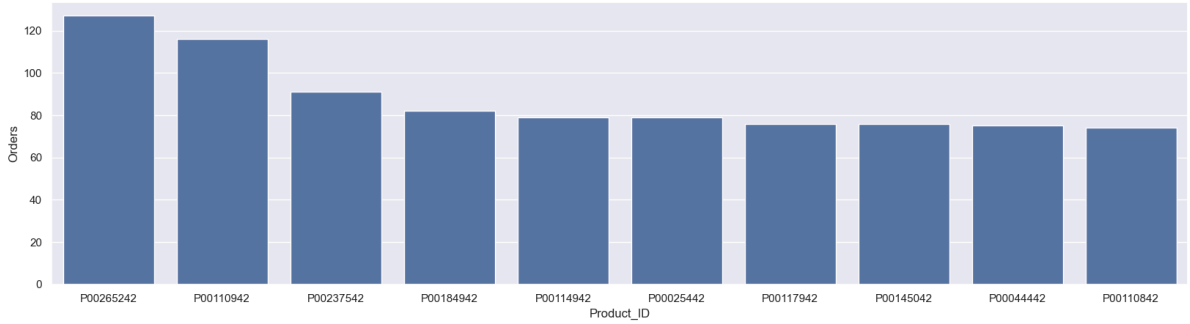


From above graphs we can see that most of the sold product ar from food, Clothing and Electronics category

In [ ]:
```
```

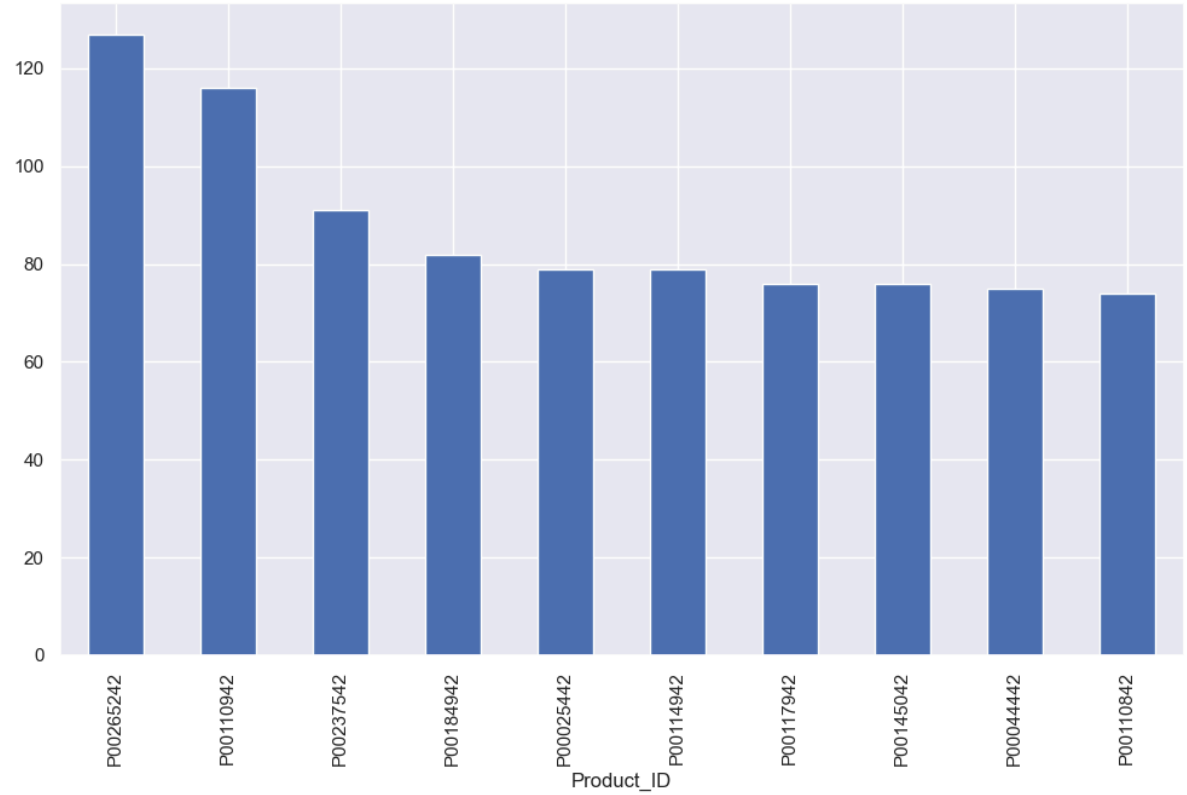
In [40]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_

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

Out[40]: <Axes: xlabel='Product_ID', ylabel='Orders'>



In [41]:
```
# 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=Fa
```

Out[41]: <Axes: xlabel='Product_ID'>



Conclusion

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

In [ ]: