In [1]:
```
import numpy as np ## array ke upper kamm kareneke liye numerical work ke liye
import pandas as pd ## dataframe ke upper work kerne ke liye table ke upper bhi
import matplotlib.pyplot as plt ## visualization data
%matplotlib inline
import seaborn as sns ## for making chart and graph
```

In [2]:
```
df=pd.read_csv("Diwali Sales Data.csv",encoding='unicode_escape')
df
```

Out[2]:

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

11251 rows × 15 columns

In [3]:
```
df.shape
```

Out[3]:
```
(11251, 15)
```

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

Out[4]:

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

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

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupatic
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemic
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthca
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Texti
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agricultu
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthca

In [6]:
```
df.describe()
```

Out[6]:

	User_ID	Age	Marital_Status	Orders	Amount	Status	unnamed1
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	0.0
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858	NaN	NaN
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869	NaN	NaN
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	NaN
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000	NaN	NaN
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000	NaN	NaN
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000	NaN	NaN
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	NaN

In [7]:
```
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 [8]:
```
df.columns
```

Out[8]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [9]:
```
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 13 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
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [11]:
```
df.isnull()
```

Out[11]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Produ
0	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	
11246	False	False	False	False	False	False	False	False	False	False	
11247	False	False	False	False	False	False	False	False	False	False	
11248	False	False	False	False	False	False	False	False	False	False	
11249	False	False	False	False	False	False	False	False	False	False	
11250	False	False	False	False	False	False	False	False	False	False	

11251 rows × 13 columns

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
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]:
```
df.dropna(inplace=True)
```

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
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              0
dtype: int64
```

In [15]:
```
df.shape
```

Out[15]:
```
(11239, 13)
```

In [16]:
```
df
```

Out[16]:

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

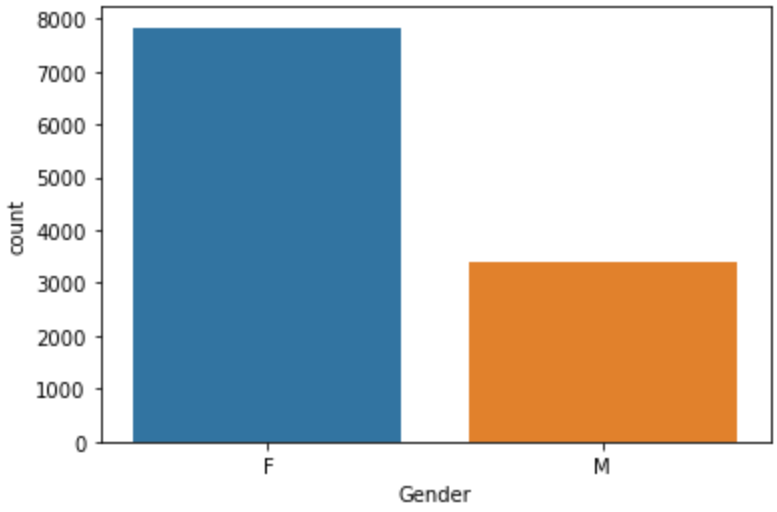
11239 rows × 13 columns

Exploratory data analysis(EDA)

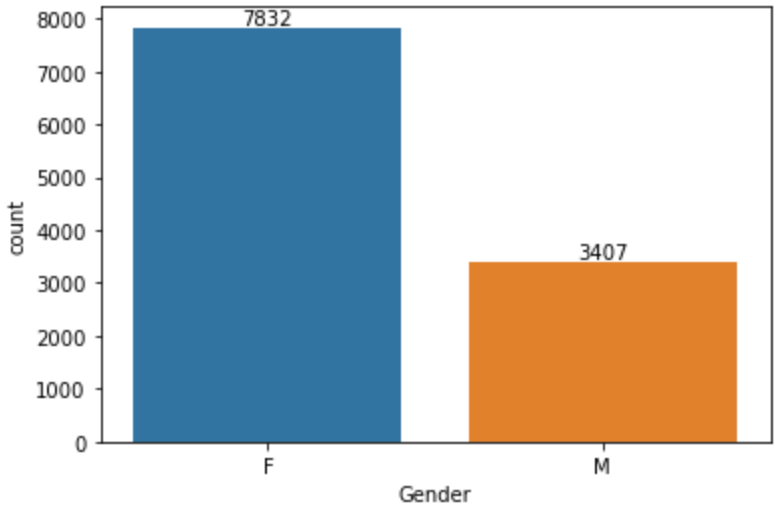
In [17]:
```
df.columns
```

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]:
```
dt=sns.countplot(x='Gender',data=df)
```



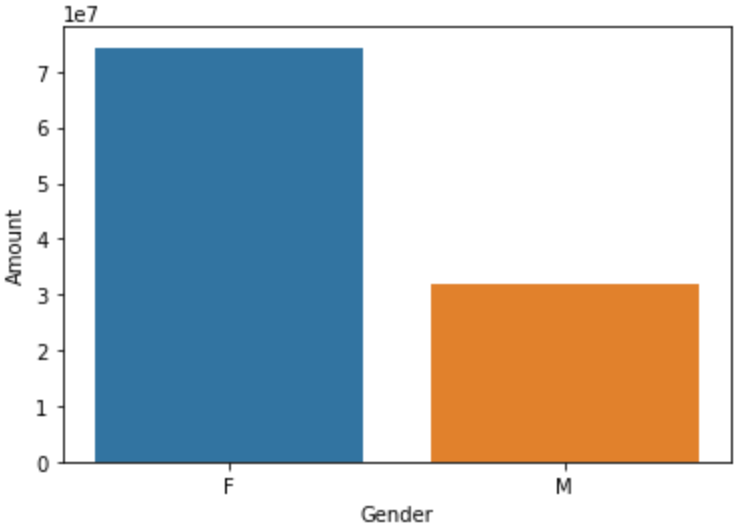
In [19]:
```
dt=sns.countplot(x='Gender',data=df)
for bar in dt.containers:
    dt.bar_label(bar)
```



In [20]:
```
purchasing_power=df.groupby(['Gender'],as_index=False)['Amount'].sum()
```

In [21]:
```
sns.barplot(x='Gender',y='Amount',data=purchasing_power)
```

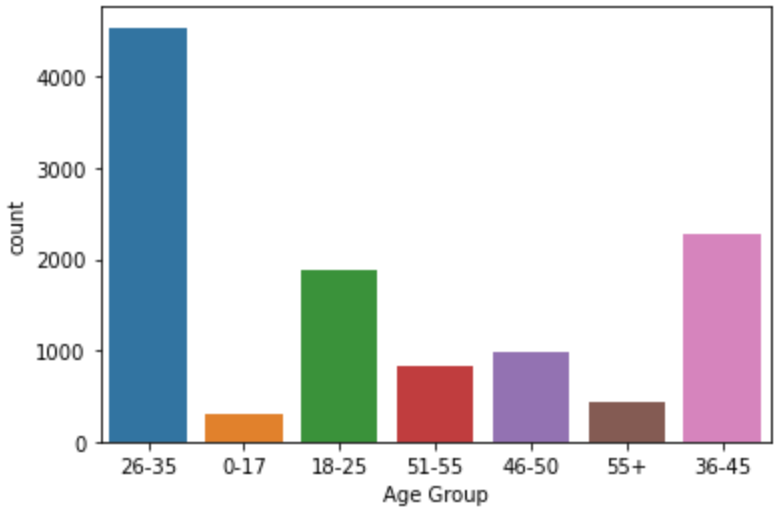
Out[21]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```



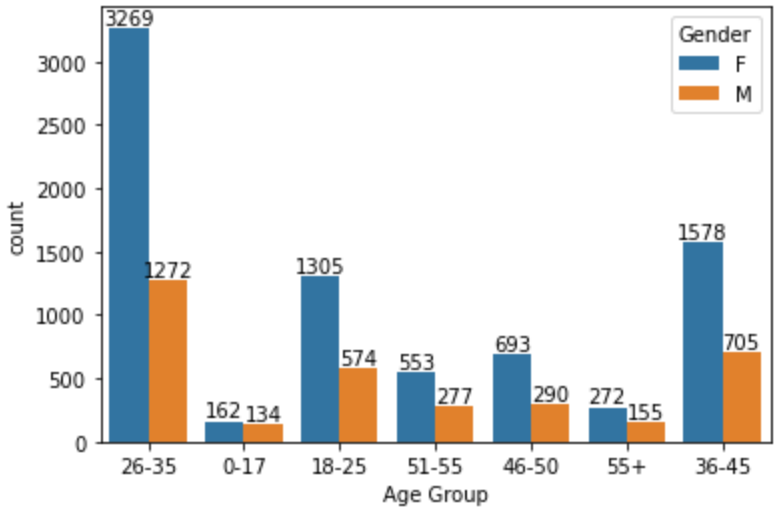
from the above graph we know the most buyer is femal even purchasing power is also greater then male

In [22]:
```
sns.countplot(x='Age Group',data=df)
```

Out[22]:
```
<AxesSubplot:xlabel='Age Group', ylabel='count'>
```

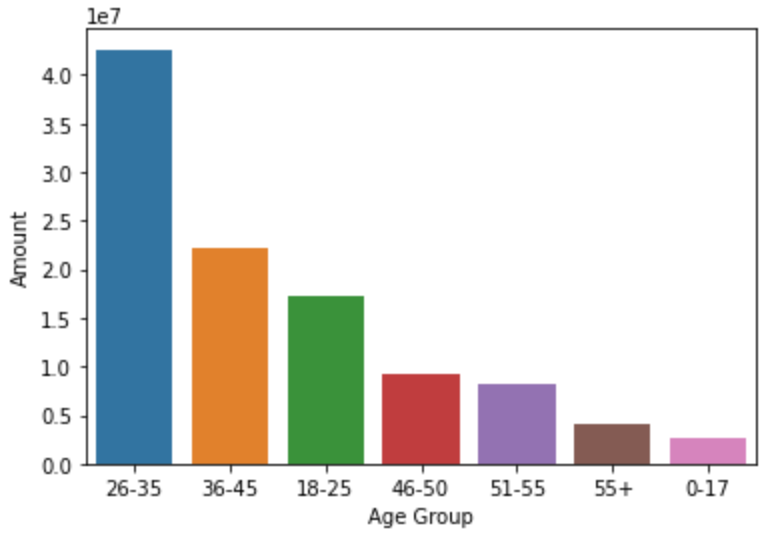


In [23]:
```
dt=sns.countplot(x='Age Group',data=df,hue='Gender')## hue ye batata hai ki kitne male o
for bars in dt.containers:
    dt.bar_label(bars)
```



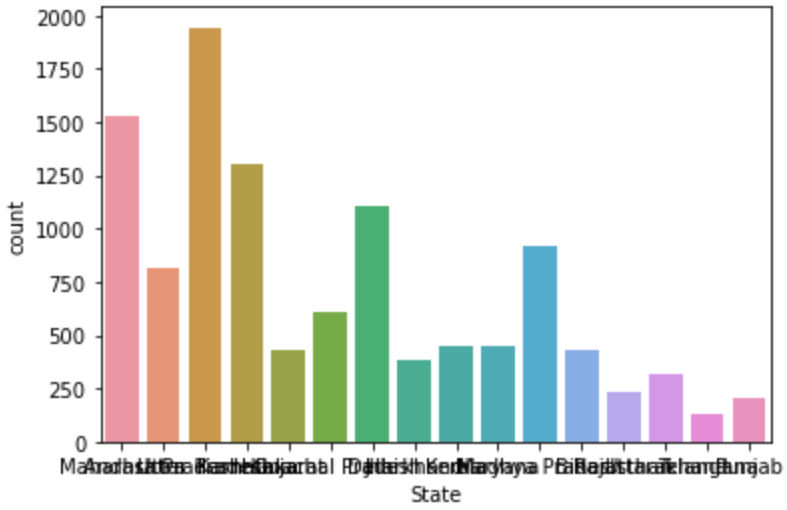
In [24]:
```
age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by='Amount',asc
sns.barplot(x='Age Group',y='Amount',data=age)
```

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

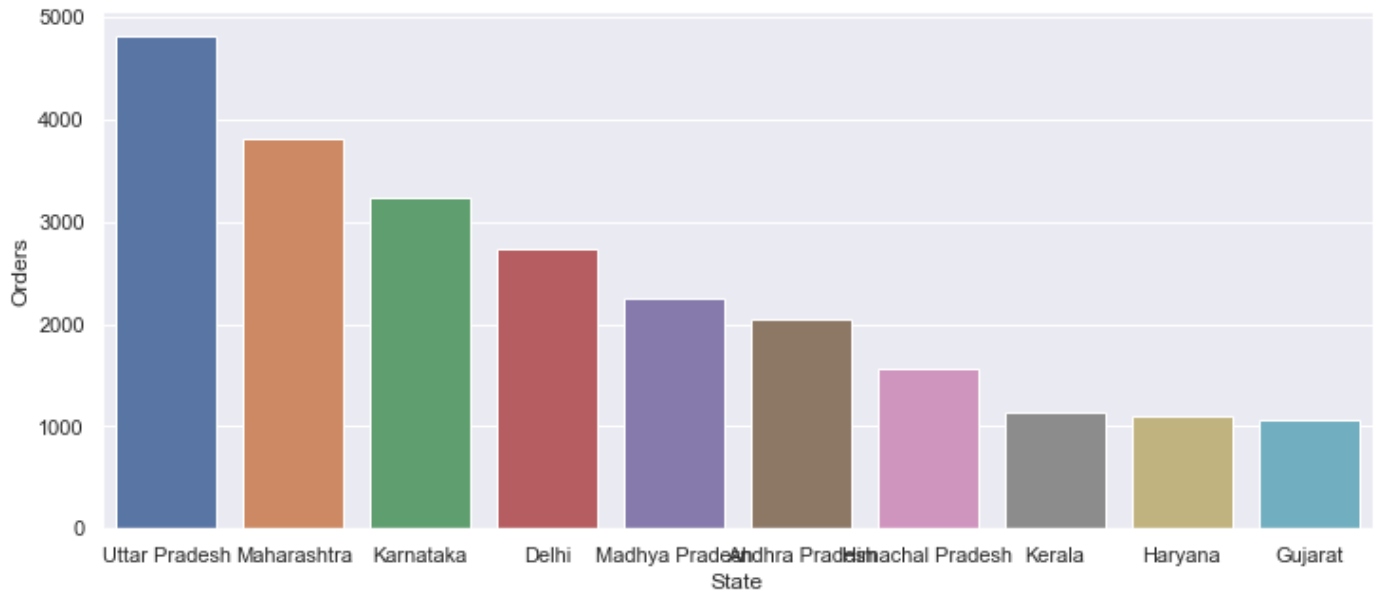


from above graph we should know that most of the buyer is from age between 26 to 35

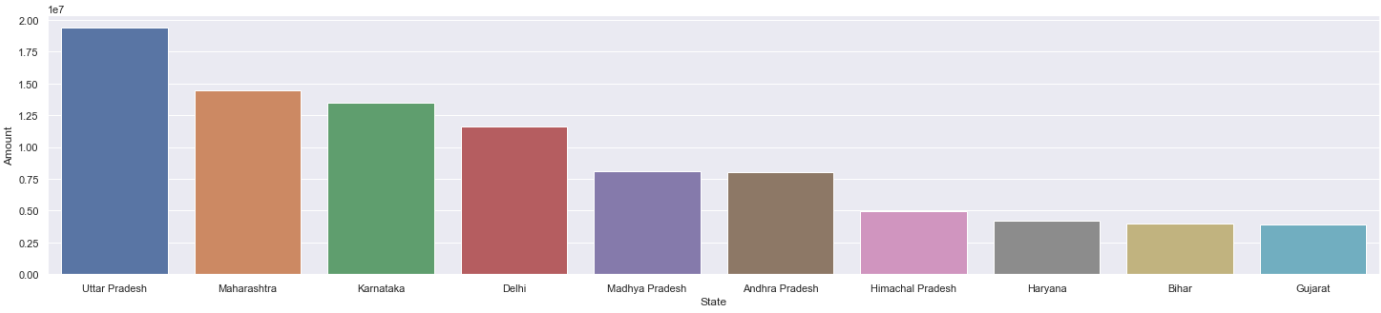
In [25]:
```
dt=sns.countplot(x='State',data=df)
```



In [32]:
```
dt=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascendin
sns.barplot(x='State',y='Orders',data=dt)
sns.set(rc={'figure.figsize':(18,5)})
```



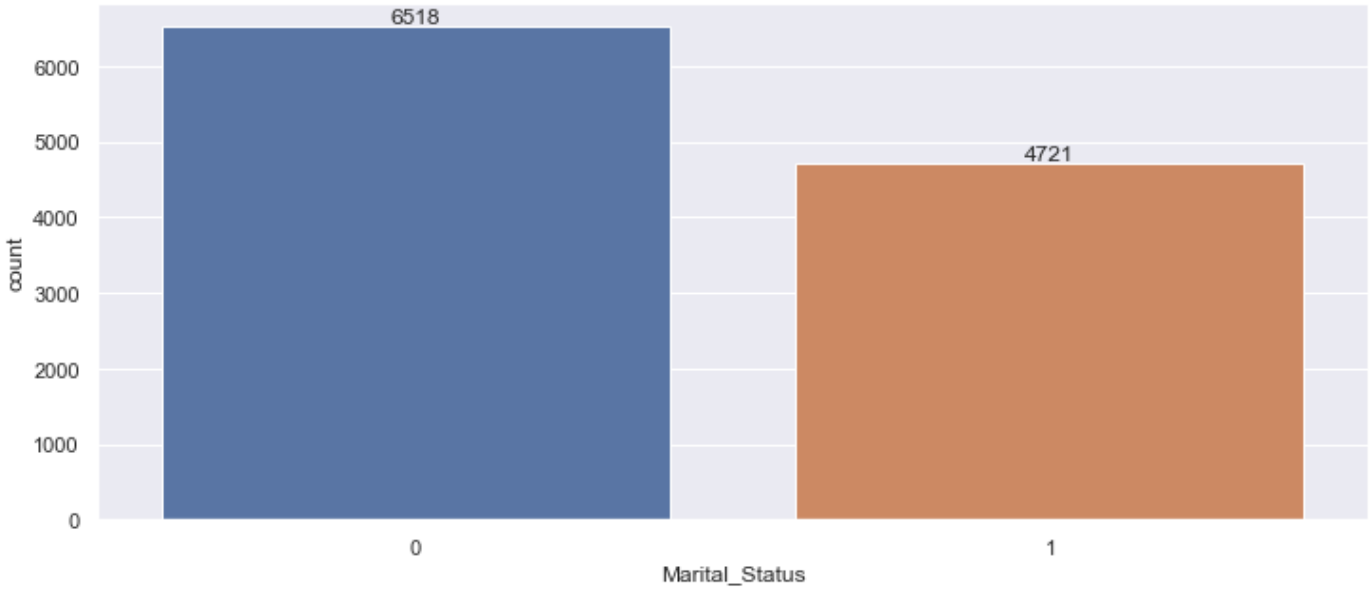
In [27]:
```
dt=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascendin
sns.barplot(x='State',y='Amount',data=dt)
sns.set(rc={'figure.figsize':(25,5)})
```



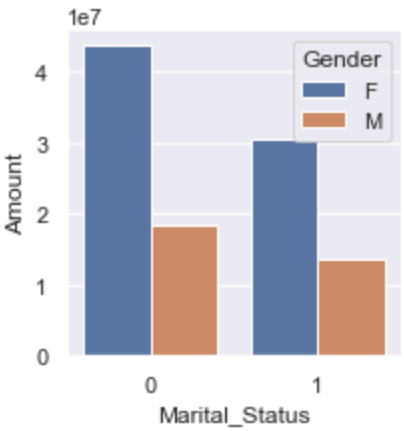
In [28]:
```
df.columns
```

Out[28]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [34]:
```
dt=sns.countplot(x='Marital_Status',data=df)
sns.set(rc={'figure.figsize':(2,5)})
for bar in dt.containers:
    dt.bar_label(bar)
```



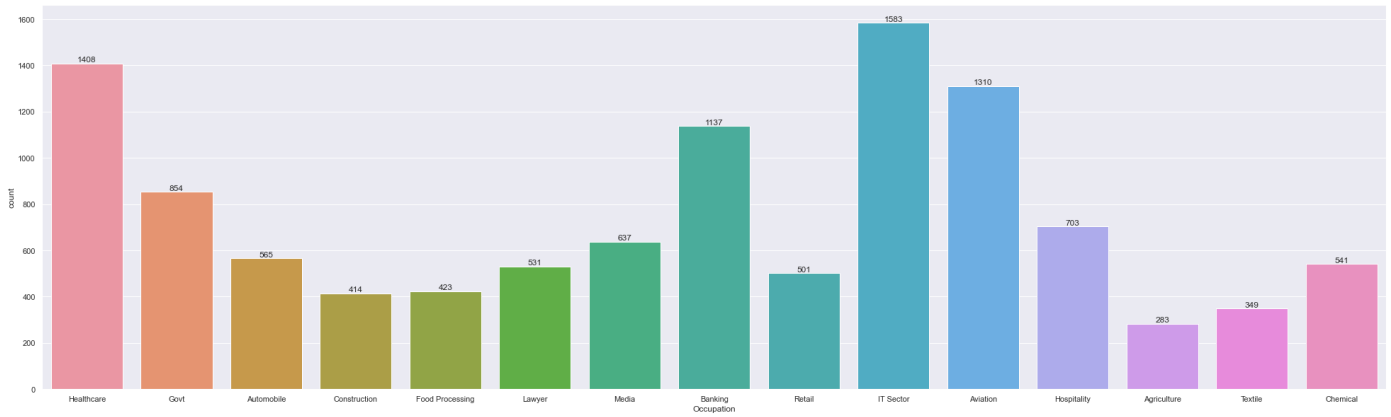
In [46]:
```
m=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by=
sns.barplot(x='Marital_Status',y='Amount',data=m,hue='Gender')
sns.set(rc={'figure.figsize':(6,3)})
```



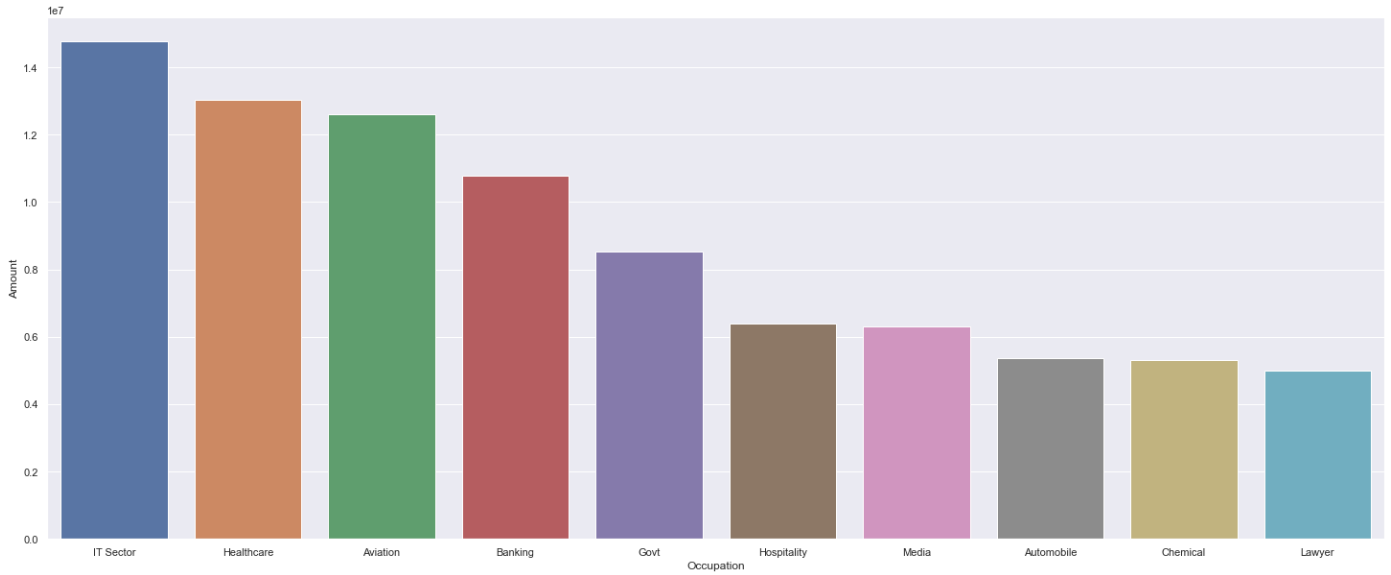
In [47]:
```
df.columns
```

Out[47]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

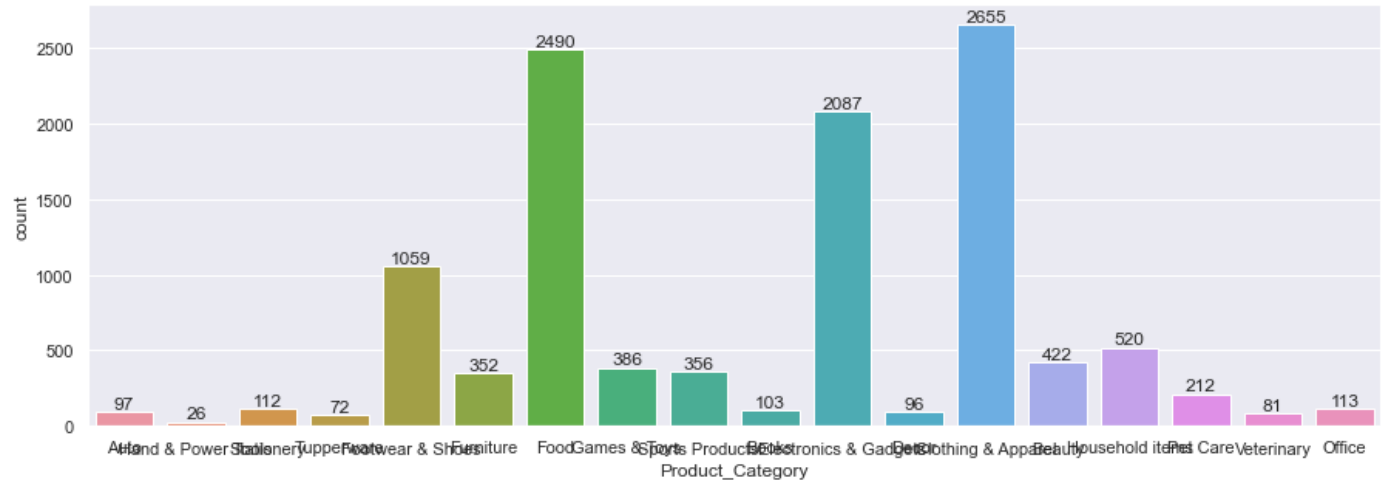
In [54]:
```
o=sns.countplot(x='Occupation',data=df)
sns.set(rc={'figure.figsize':(25,10)})
for bar in o.containers:
    o.bar_label(bar)
```



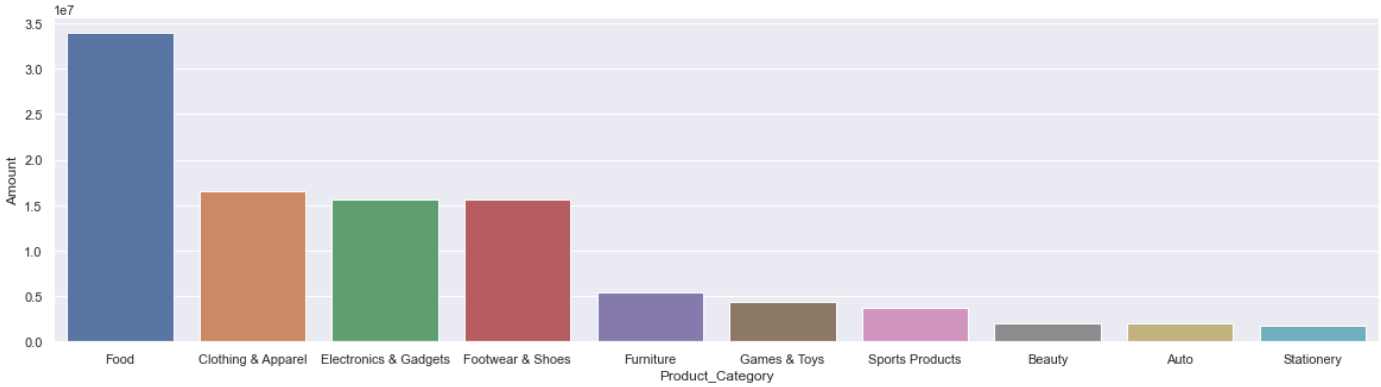
In [55]:
```
o=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',asce
sns.barplot(x='Occupation',y='Amount',data=o)
sns.set(rc={'figure.figsize':(25,10)})
```



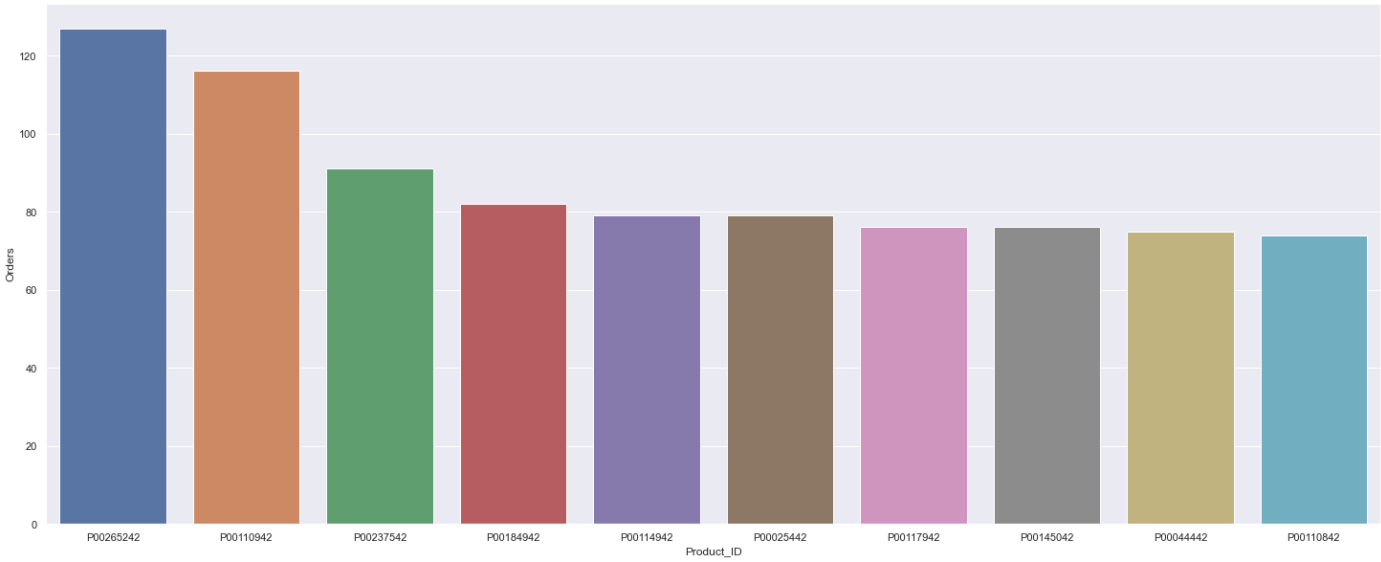
In [63]:
```
p=sns.countplot(x='Product_Category',data=df)
for bar in p.containers:
    p.bar_label(bar)
sns.set(rc={'figure.figsize':(20,5)})
```



In [64]:
```
p=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by='Amount
sns.barplot(x='Product_Category',y='Amount',data=p)
sns.set(rc={'figure.figsize':(25,10)})
```



In [68]:
```
a=sns.countplot(x='Product_ID',data=df)
o=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='Orders',asce
sns.barplot(x='Product_ID',y='Orders',data=o)
sns.set(rc={'figure.figsize':(25,10)})
```



conclusion:

Married women age group 25-35yrs UP,Maharshtra,Kernataka working in IT sector and heathcare are more likely to buy product from food ,cloths and electric category

In [ ]: