In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df = pd.read_excel('C:/Users/Suraj Soni/Desktop/Diwali Sales Data.xlsx')
```

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

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

In [4]:
```
df.head(3)
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zon
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Westerr
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southerr
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Centra

In [5]:
```
df.tail(3)
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zo
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Cent
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southe
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Weste

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 [7]:
```
#dropping blank coloumns
df.drop(['Status','unnamed1'],axis=1, inplace = True)
```

In [8]:
```
#checking for null value
pd.isnull(df)
```

Out[8]:

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

11251 rows × 13 columns

In [9]:
```
pd.isnull(df).sum()
```

Out[9]:
```
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 [10]:
```
#droping null
df.dropna(inplace=True)
```

In [11]:
```
df.shape         #12 null values droped
```

Out[11]:
```
(11239, 13)
```

In [12]:
```
#changing the data type of Amount to Int
df['Amount'] = df['Amount'].astype('int')
```

In [13]:
```
df['Amount'].dtype
```

Out[13]:
```
dtype('int32')
```

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

Out[14]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [15]:
```
df.describe()
```

Out[15]:

	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 [16]:
```
#applying discribe function on selected coloumns.
df[['Age','Orders','Amount']].describe()
```

Out[16]:

	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

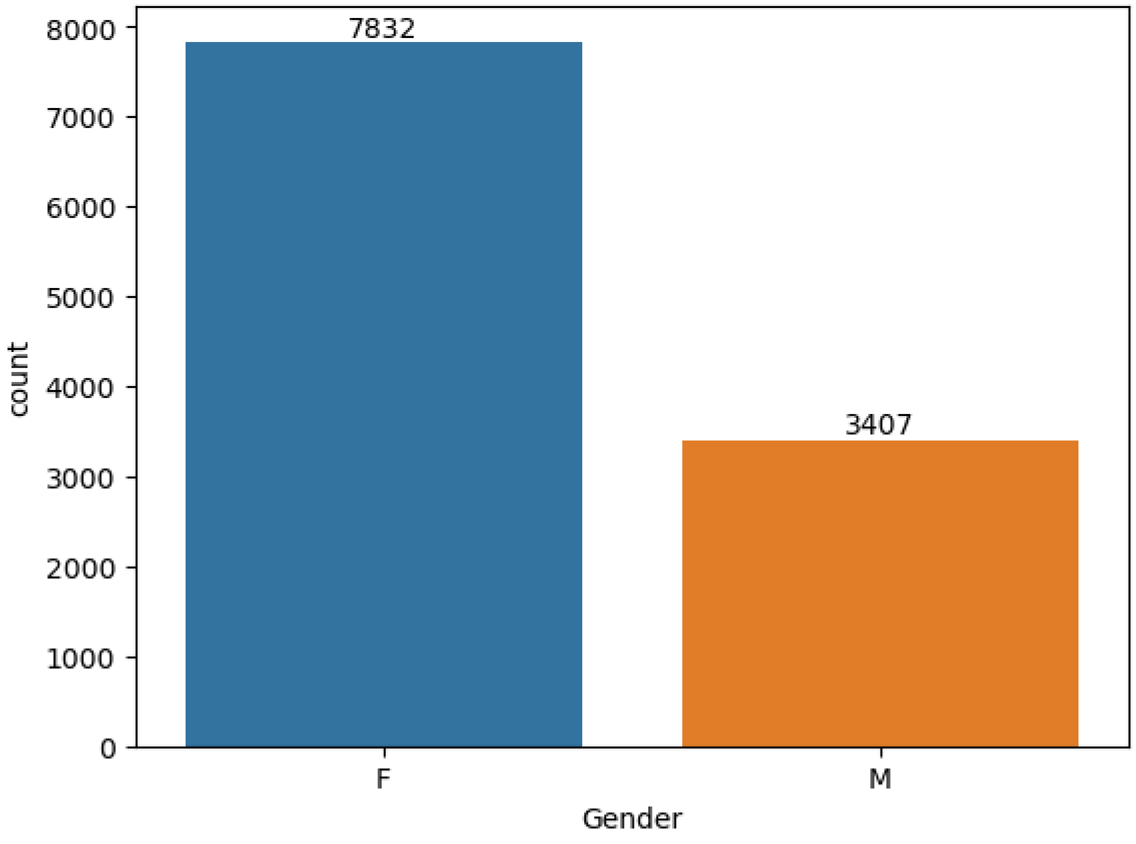
EDA

Gender

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

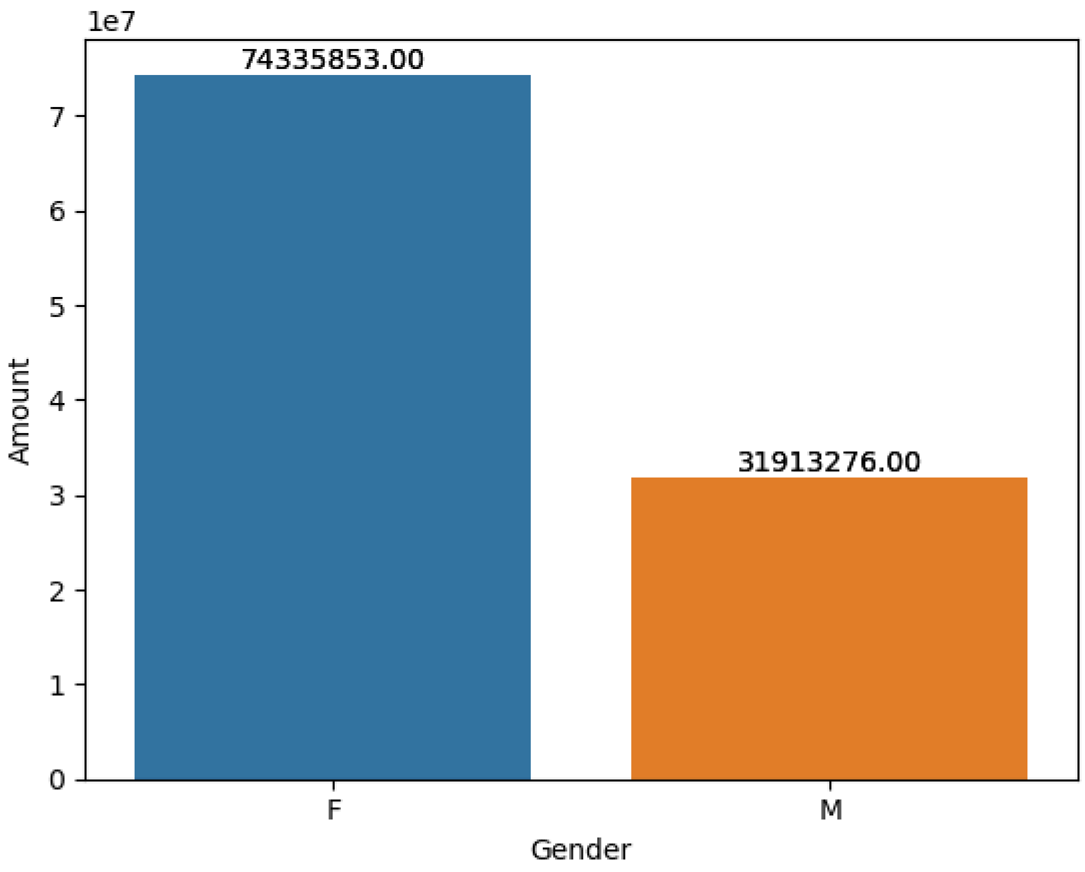


In [19]:
```
df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', asc
```

Out[19]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [20]:
```
sales = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amou

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

ax = sns.barplot(x='Gender', y='Amount', data=sales)

# Add labels to the bars
for container in ax.containers:
    ax.bar_label(container, fmt='%.2f')
```



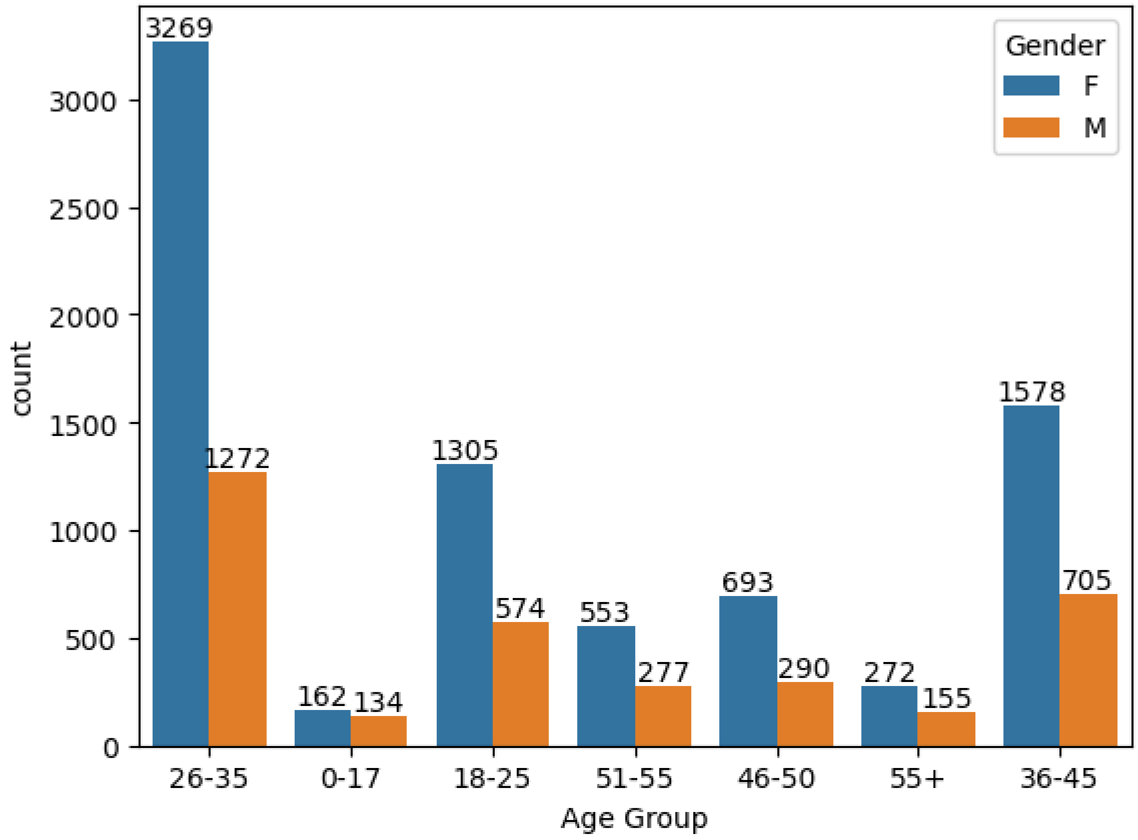
Females have high orders

Age

In [21]:
```
df.columns
```

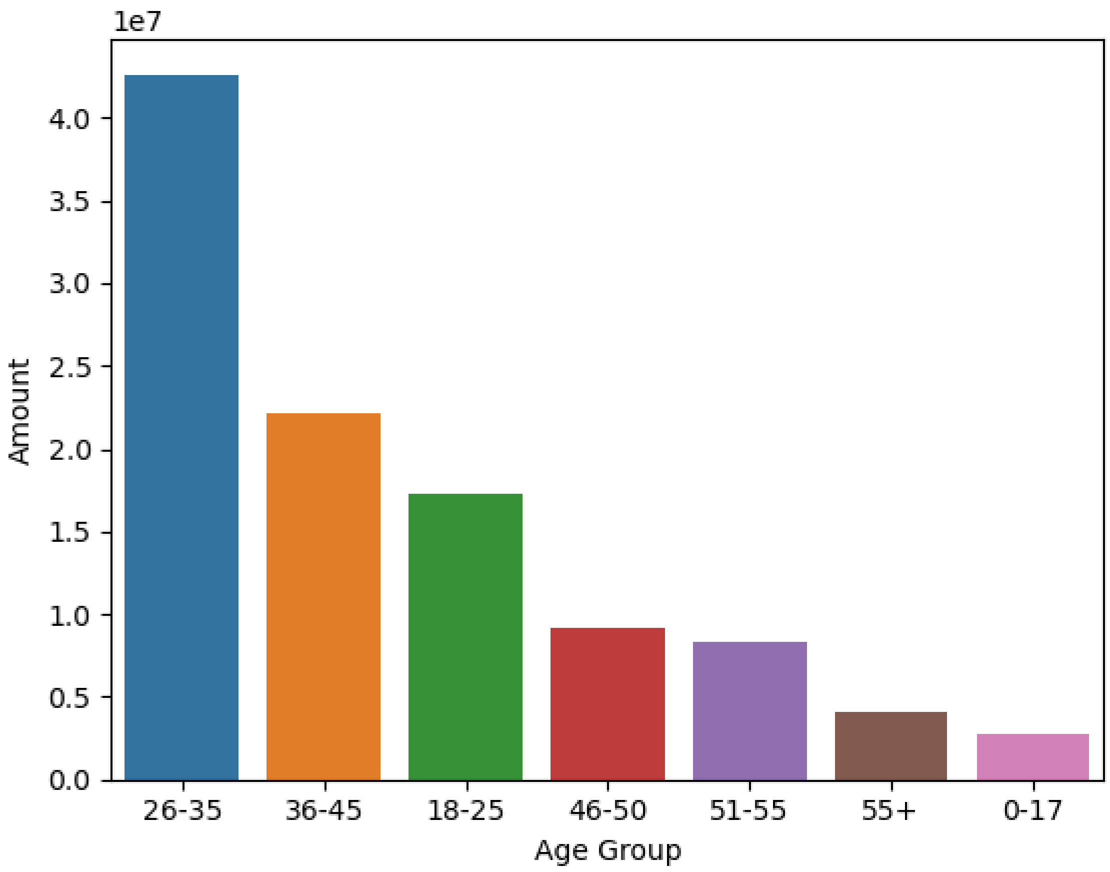
Out[21]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [22]:
```
ax = sns.countplot(x='Age Group',data = df,hue='Gender')
for container in ax.containers:
    ax.bar_label(container)
```



```
In [23]: sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(b

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

```
Out[23]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



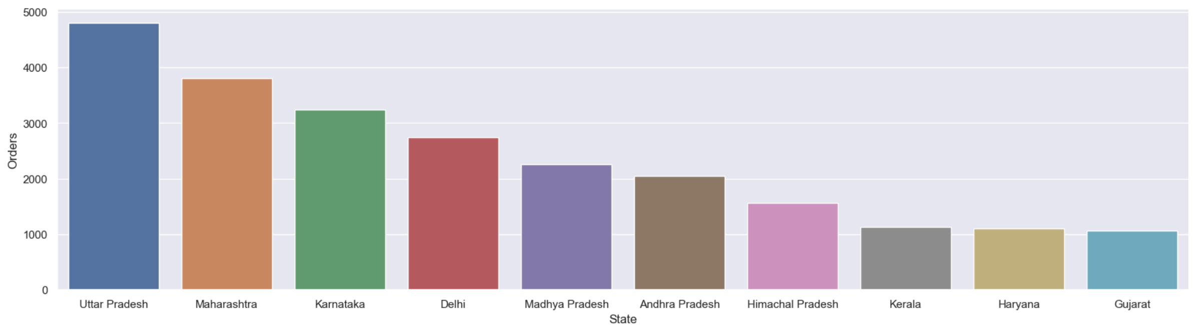
Most of the buyer are of are group from 26 to 35

In [24]:
```
df.columns
```

Out[24]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [35]:
```
#total number of order from top
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by=
sns.set(rc={'figure.figsize':(20,5)})

sns.barplot(x='State',y='Orders',data=sales_state)
```

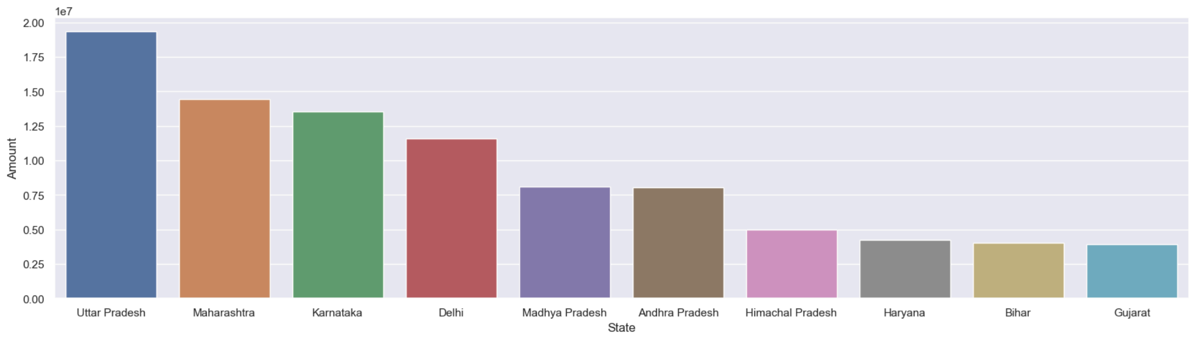
Out[35]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



from the above analysis we can find that UP, Maharastra and Karnataka are the top 3 state in terms of highest orders.

In [34]:
```
#total amount/sales from top 10 state
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by=
sns.set(rc={'figure.figsize':(20,5)})

sns.barplot(x='State',y='Amount',data=sales_state)
```

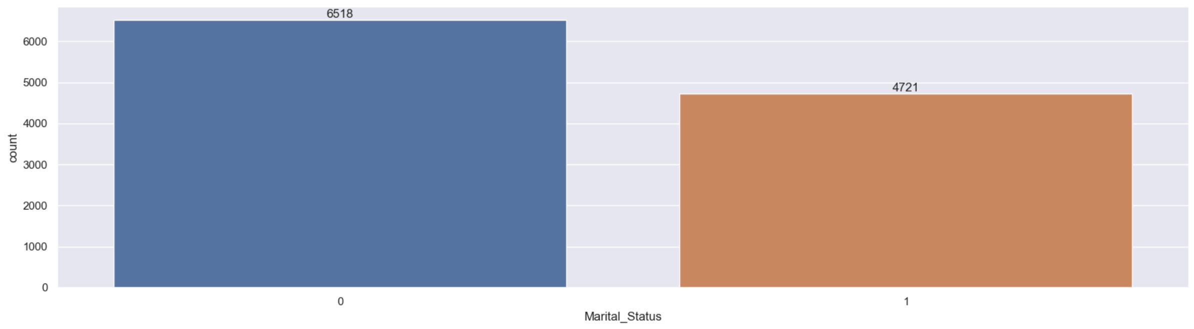
Out[34]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



from the above analysis we can find that UP, Maharastra and Karnataka are the top 3 state in maximum revenue.

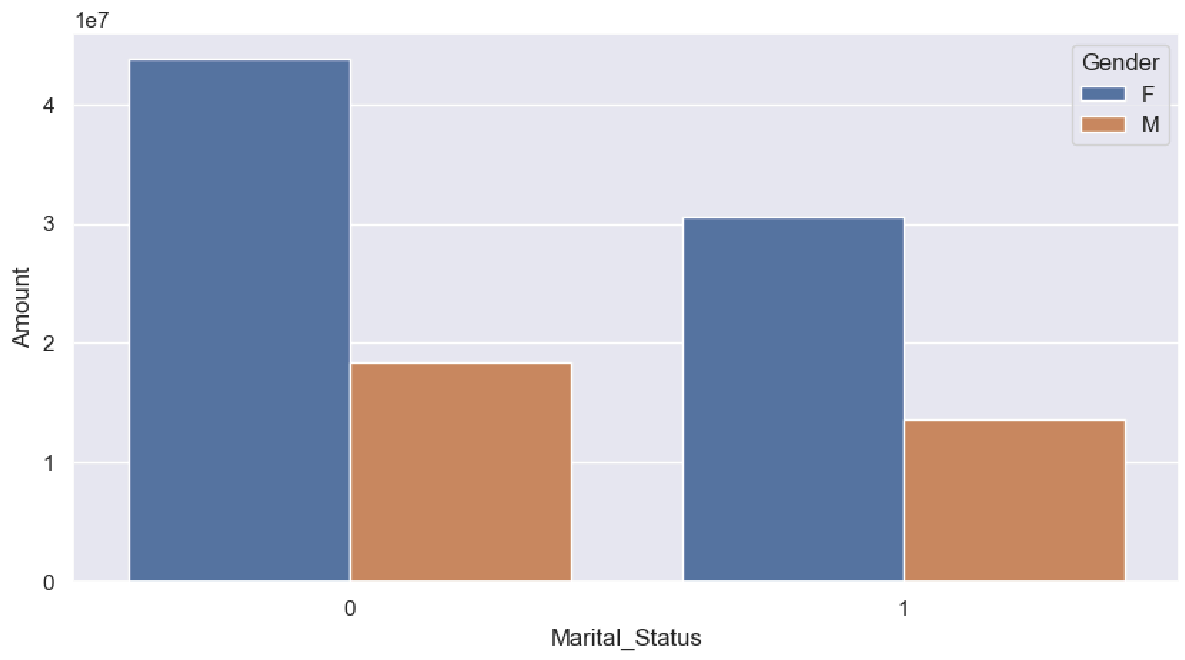
Merital Status

In [37]:
```
ax = sns.countplot(x='Marital_Status',data=df)

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



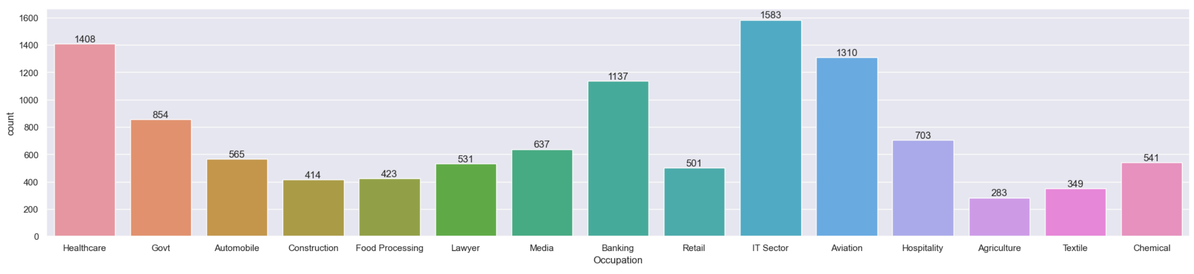
In [40]:
```
sales_state = df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].sum
sns.set(rc={'figure.figsize':(10,5)})

sns.barplot(x='Marital_Status',y='Amount',data=sales_state,hue = 'Gender')
```

Out[40]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



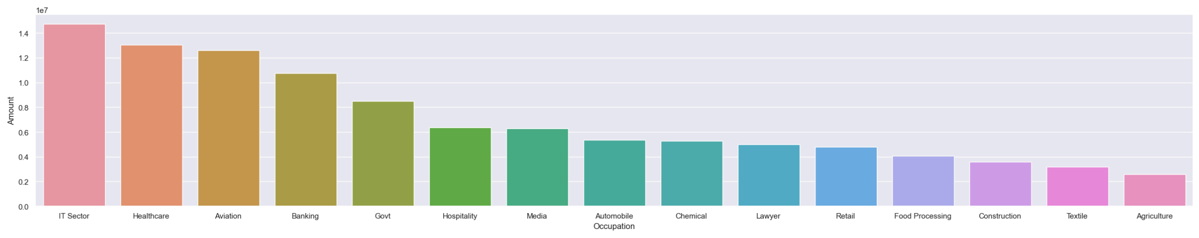
Occupation

In [44]:
```
ax = sns.countplot(x='Occupation',data=df)
sns.set(rc={'figure.figsize':(30,5)})
for container in ax.containers:
    ax.bar_label(container)
```



In [54]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_value
sns.set(rc={'figure.figsize':(30,5)})

sns.barplot(x='Occupation',y='Amount',data=sales_state)
```

Out[54]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

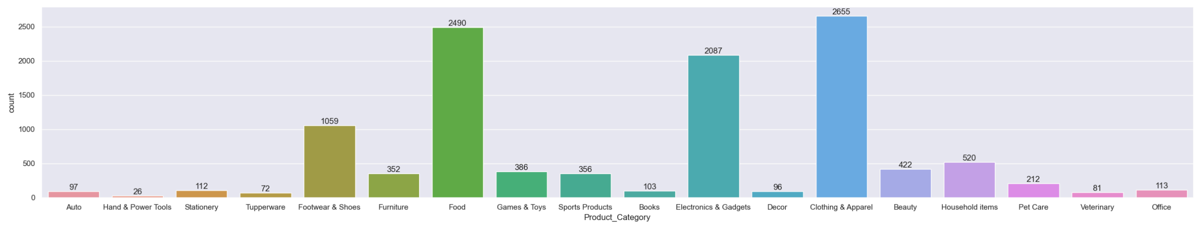


```
In [48]: df.columns
```

```
Out[48]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
                'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
                'Orders', 'Amount'],
               dtype='object')
```

Product_Category

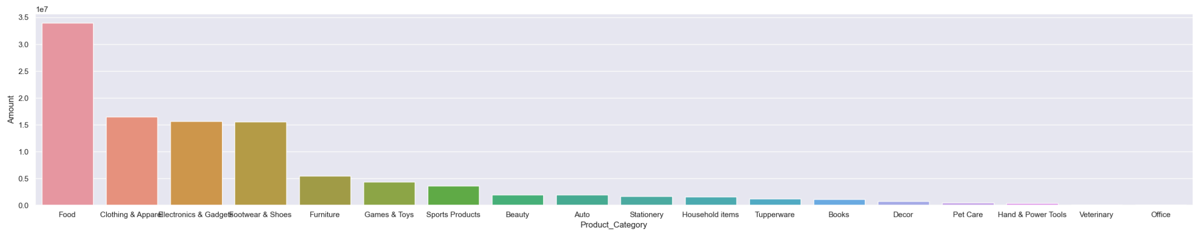
```
In [52]: ax = sns.countplot(x='Product_Category',data=df)

         sns.set(rc={'figure.figsize':(25,5)})
         for container in ax.containers:
             ax.bar_label(container)
```



Total Number of order received from clothing, food and electronics.

```
In [55]: sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort
         sns.set(rc={'figure.figsize':(30,5)})

         sns.barplot(x='Product_Category',y='Amount',data=sales_state)
```

```
Out[55]: <Axes: xlabel='Product_Category', ylabel='Amount'>
```

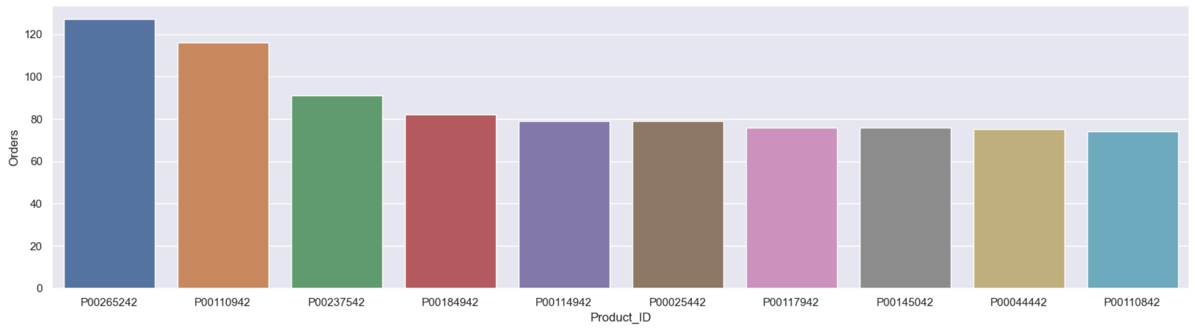


Food has the highest revenew followed by clothing and electronics.

Product ID

```
In [60]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_value
         sns.set(rc={'figure.figsize':(20,5)})

         sns.barplot(x='Product_ID',y='Orders',data=sales_state)
```

```
Out[60]: <Axes: xlabel='Product_ID', ylabel='Orders'>
```


Orders
0
20
40
60
80
100
120
P00265242
P00110942
P00237542
P00184942
P00114942
P00025442
P00117942
P00145042
P00044442
P00110842
Product_ID

```
In [ ]:
```