In [3]:
```
#importing libraries

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

In [ ]:
```
#Reading the File
```

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

In [ ]:
```
#checking the dataset dimensions
```

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

Out[5]:
```
(11251, 15)
```

In [ ]:
```
# checking the dataet columns
```

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

Out[6]:

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

In [ ]:
```
#To check for any null values
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
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
Status              11251
unnamed1            11251
dtype: int64
```

In [8]:
```
df.dtypes
```

Out[8]:
```
User_ID               int64
Cust_name            object
Product_ID           object
Gender               object
Age Group            object
Age                   int64
Marital_Status        int64
State                object
Zone                 object
Occupation           object
Product_Category     object
Orders                int64
Amount              float64
Status              float64
unnamed1            float64
dtype: object
```

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 [ ]:
```
#Deleting the unwanted columns
```

In [10]:
```
df.drop(['Status','unnamed1'],axis=1, inplace=True)
```

In [11]:
```
df
```

Out[11]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Stat
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtr
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prades
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prades
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnatak
4	1000588	Joni	P00057942	M	26-35	28	1	Gujara
...	...	...	...	...	...	...	...	.
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtr
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryan
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhy Prades
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnatak
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtr

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 [ ]:
```
#dropping the rows containing null values
```

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 [ ]:

```
#changing the datatype of the columns
```

In [15]:

```
df['Amount']=df['Amount'].astype('int')
```

In [16]:

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

Out[16]:

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

In [ ]:

```
#Check for the columns present in the dataset
```

In [17]:

```
df.columns
```

Out[17]:

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

In [18]:

```
df.describe()
```

Out[18]:

	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 [19]:

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

Out[19]:

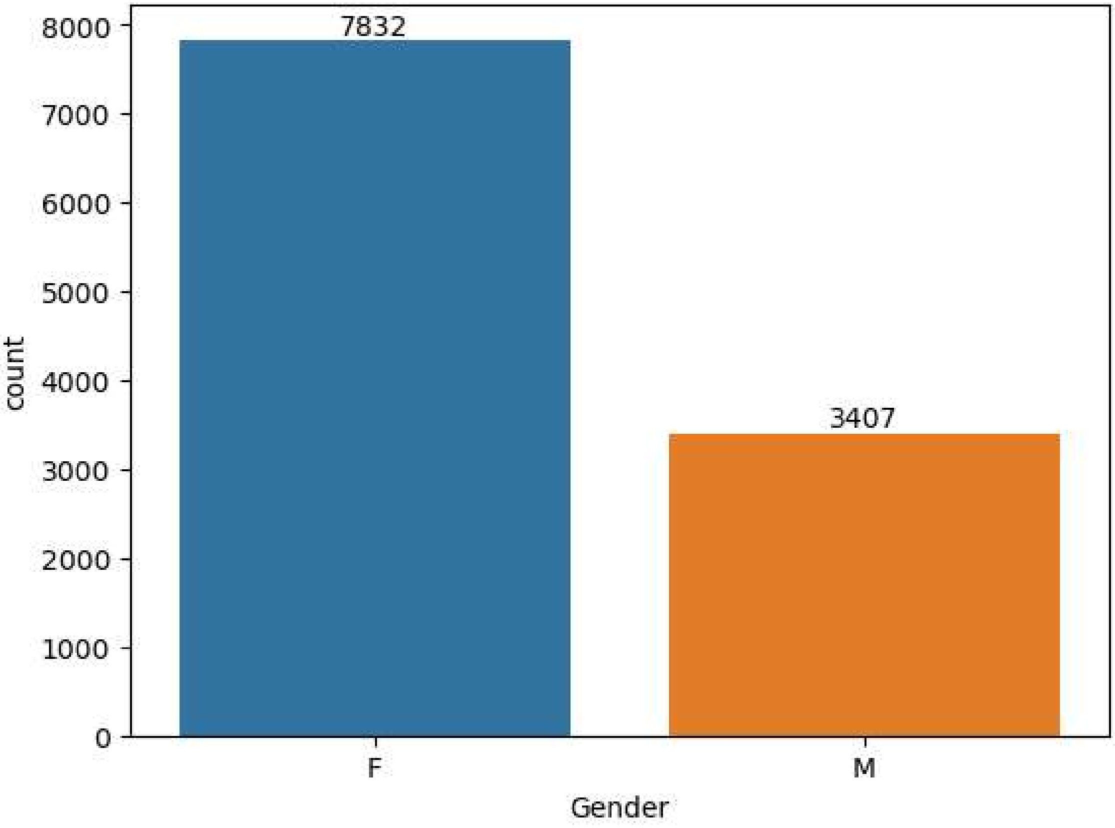
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

EDA

In [20]:

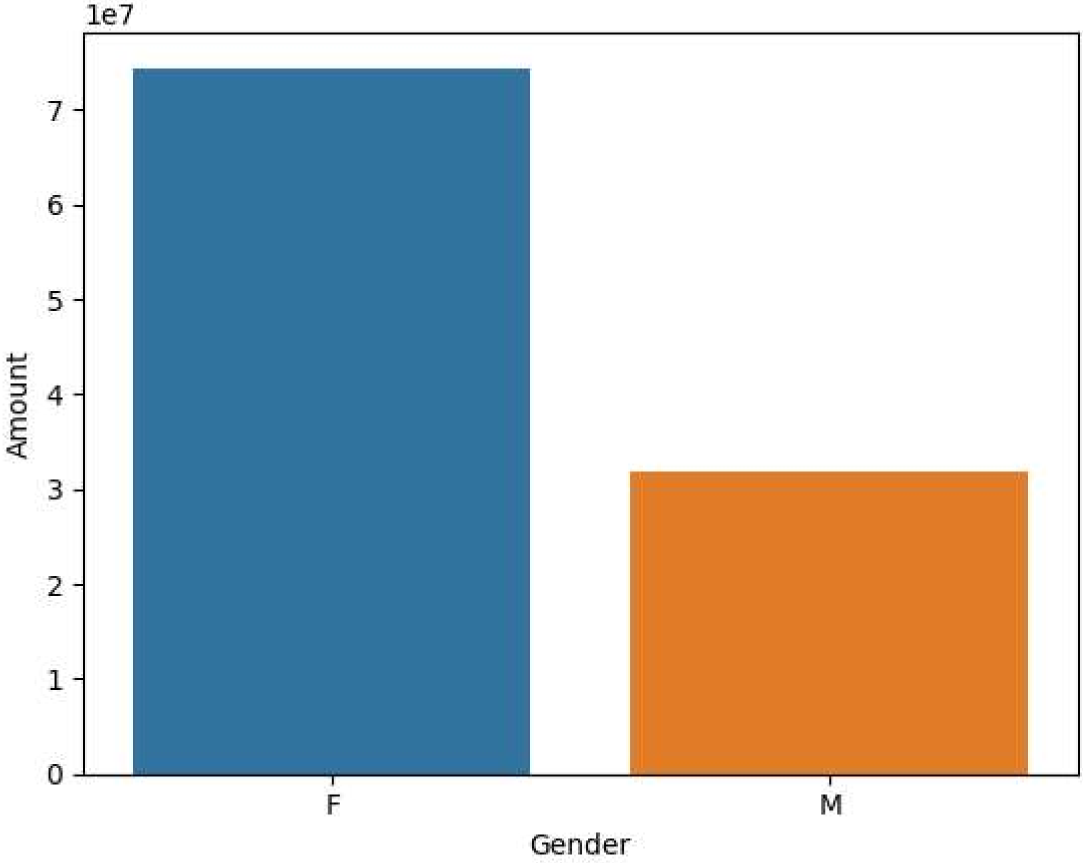
```
ax=sns.countplot(x="Gender",data=df)

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



```
In [21]: sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_valu
         sns.barplot(x = 'Gender', y= 'Amount', data=sales_gen )
```

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

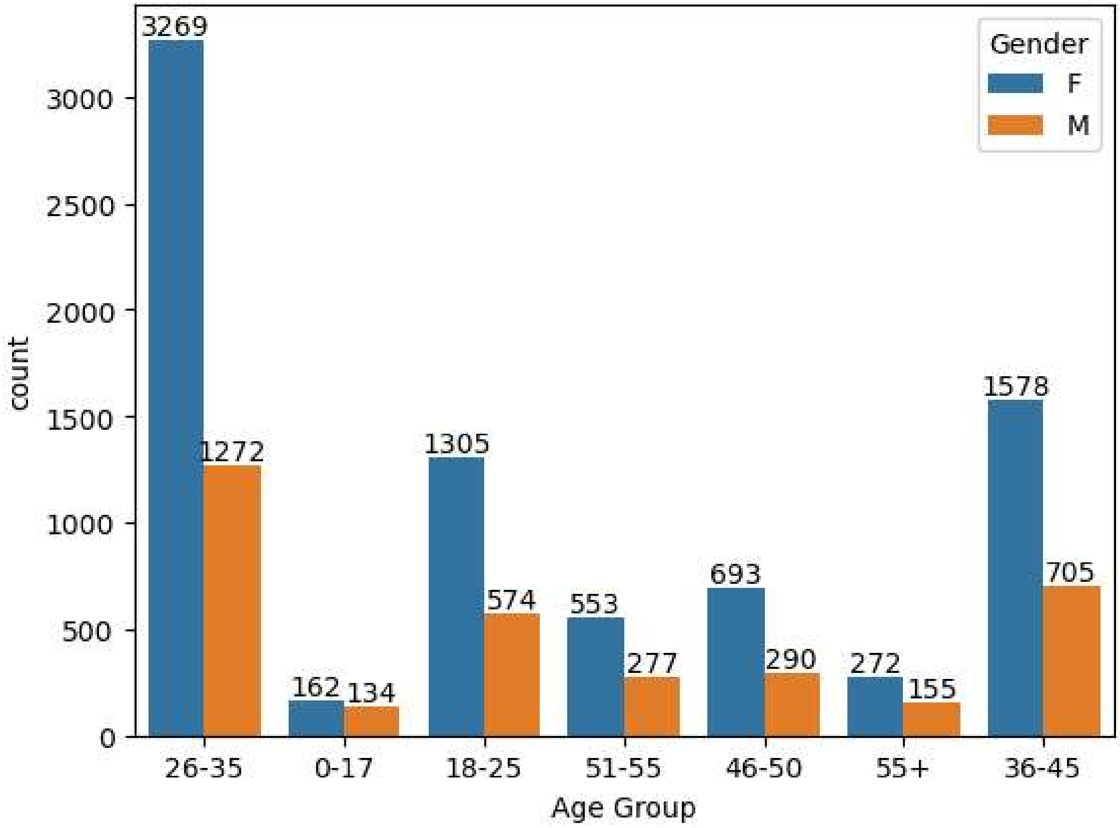


From the above two graphs we can see that most of the buyers are female and the purchasing power of the females are greater than men

In [22]:

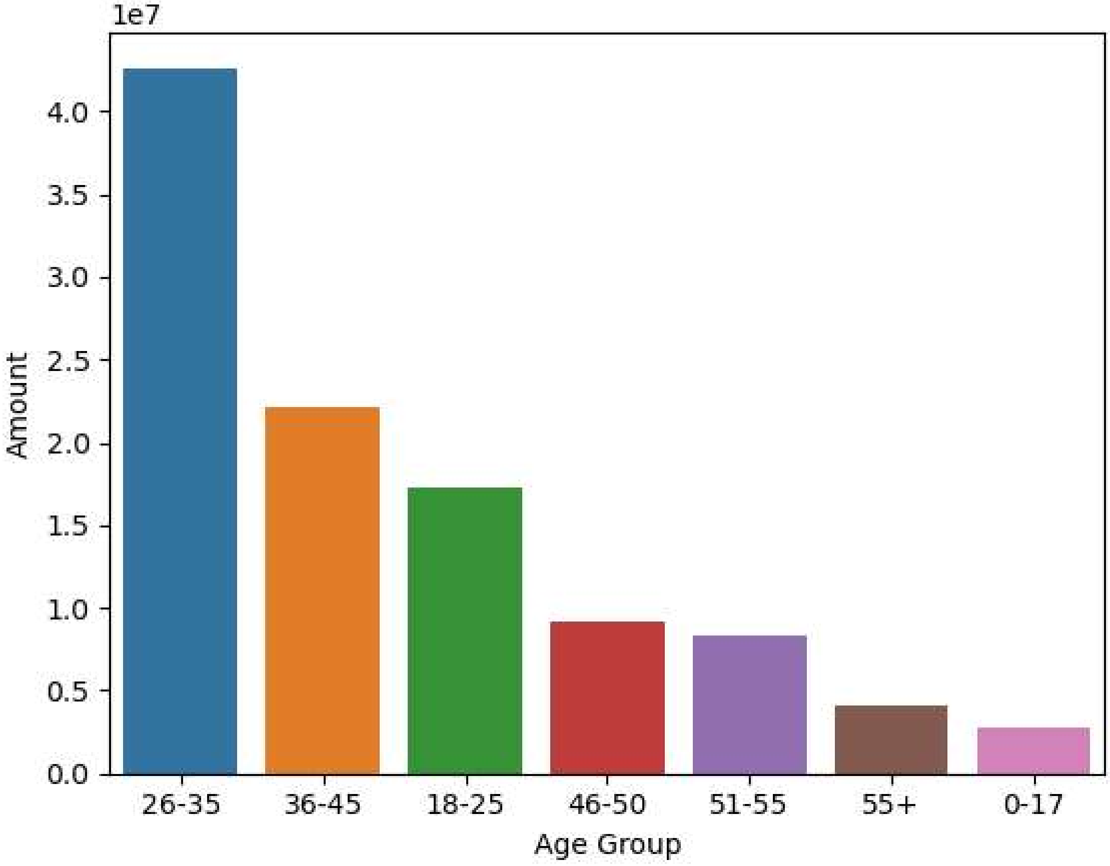
```
ax=sns.countplot(data=df, x='Age Group',hue='Gender')

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



```
In [23]: sales_age = df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_v
         sns.barplot(x = 'Age Group', y= 'Amount', data=sales_age)
```

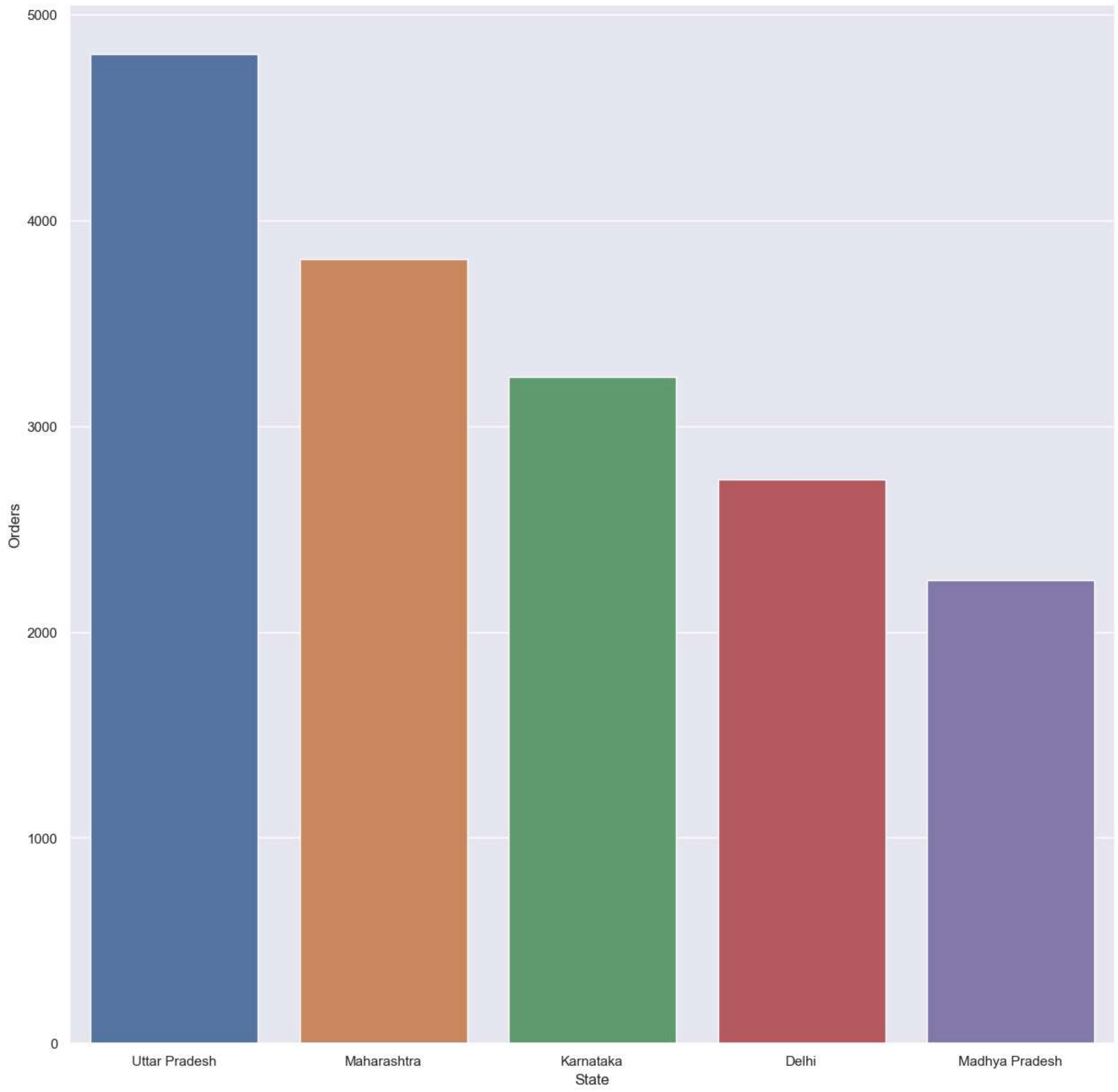
```
Out[23]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



From the above graphs we can see that people of age group 26-35 have made the maximum number of purchases as well as they hace also spent the maximum amount

In [24]:
```
sales_state = df.groupby(['State'],as_index=False)['Orders'].sum().sort_val
sns.set(rc={'figure.figsize':(15,15)})
sns.barplot(data=sales_state,x='State',y='Orders')
```

Out[24]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



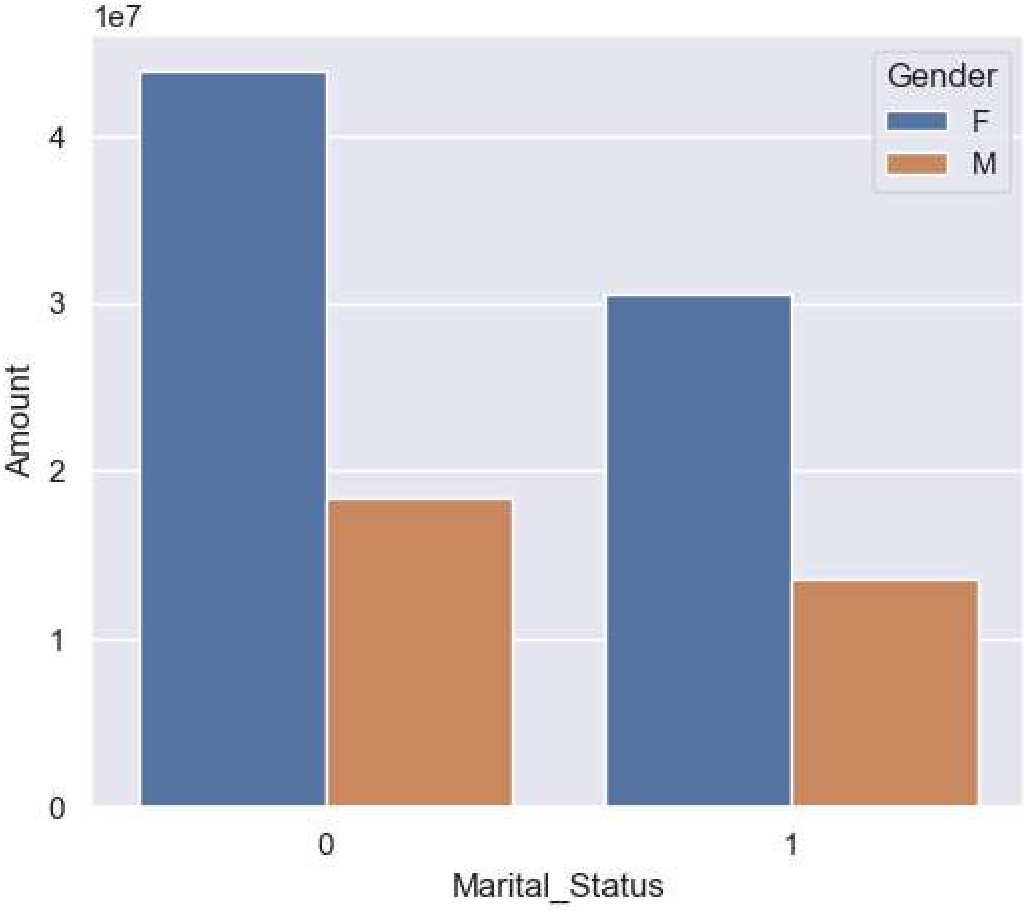
In [ ]:
```
The above graph indicated that the people of maharashtra have ordered the m
```

In [25]:

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

Out[25]:

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

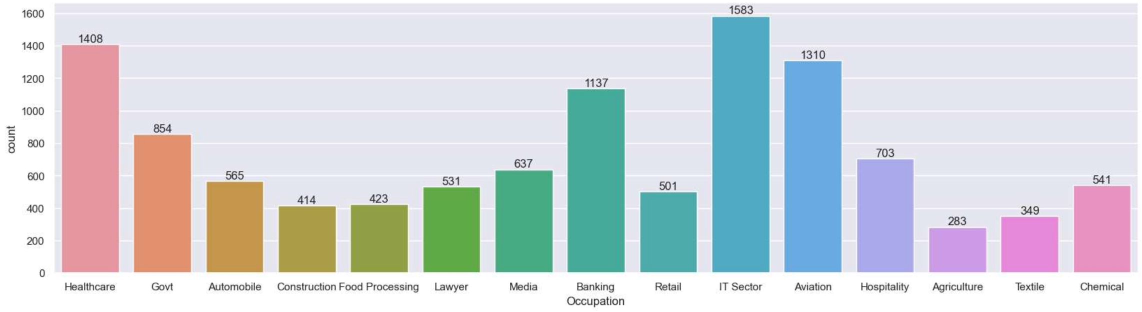


The above graph indicates that the married females have spent maximum in ordering

In [26]:

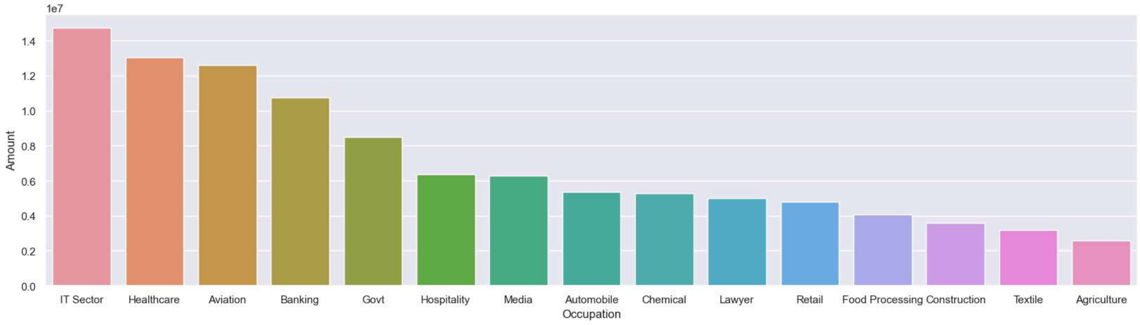
```
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data=df, x='Occupation')

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



```
In [27]: 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')
```

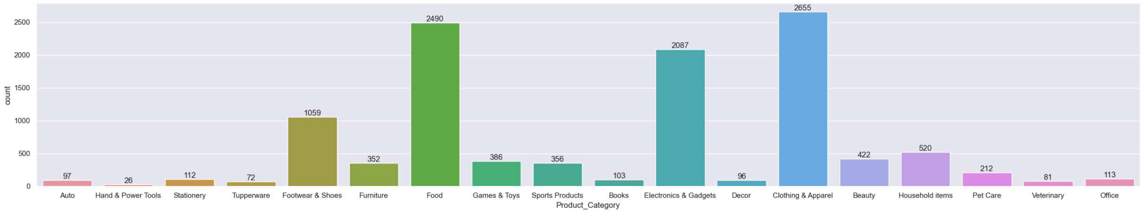
```
Out[27]: <Axes: xlabel='Occupation', ylabel='Amount'>
```



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

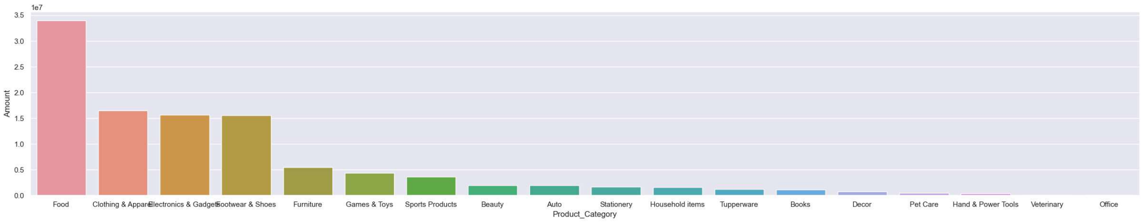
```
In [28]: sns.set(rc={'figure.figsize':(30,5)})
         ax=sns.countplot(data=df, x='Product_Category')

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



```
In [29]: sales_state=df.groupby(['Product_Category'],as_index=False)['Amount'].sum()
         sns.set(rc={'figure.figsize':(30,5)})
         sns.barplot(data=sales_state,x='Product_Category',y='Amount')
```

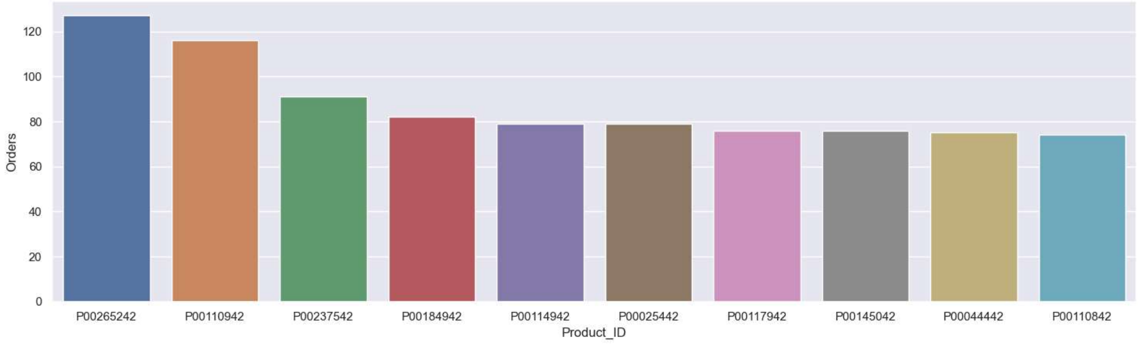
```
Out[29]: <Axes: xlabel='Product_Category', ylabel='Amount'>
```



From the above graphs we can see that the maximum number of products related to foods has been ordered as well as maximum amount has been spent on the food items.

In [30]:
```
sales_state=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_
sns.set(rc={'figure.figsize':(18,5)})
sns.barplot(data=sales_state,x='Product_ID',y='Orders')
```

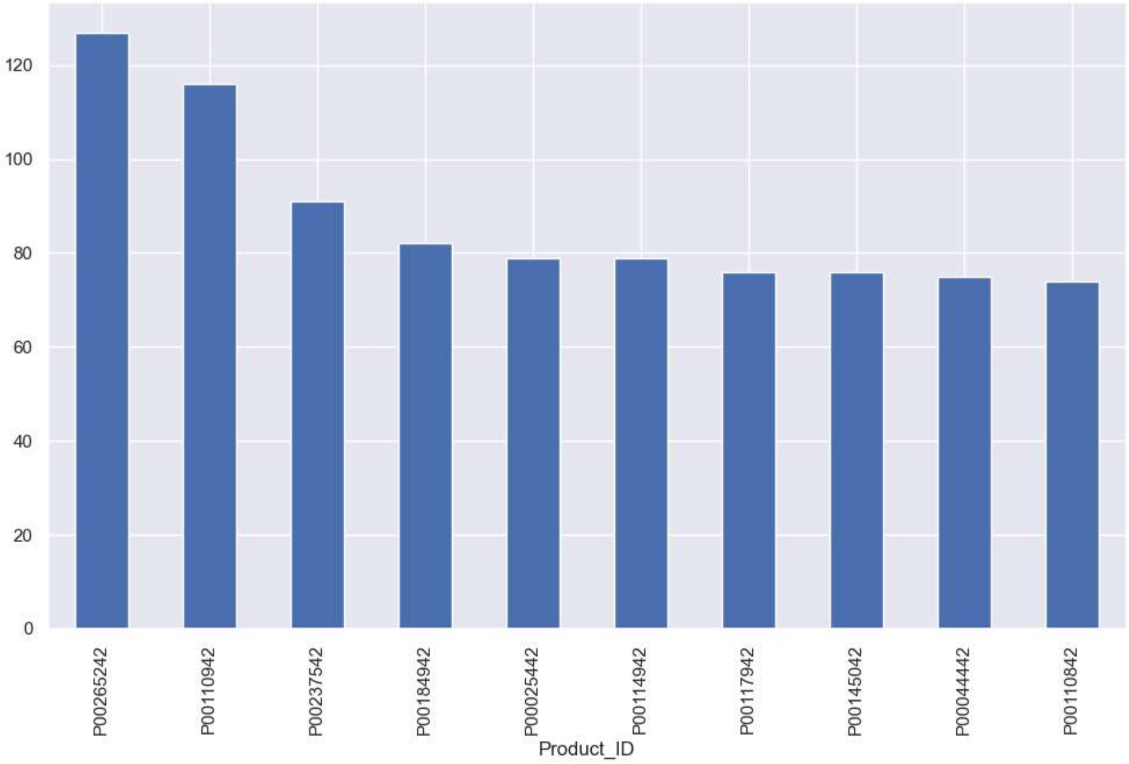
Out[30]:
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



In [31]:
```
fig1, ax1=plt.subplots(figsize=(12,7))

df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending
```

Out[31]:
```
<Axes: xlabel='Product_ID'>
```



Top 10 most sold products

Conclusion

The key findings from the EDA are that the 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.