# Diwali Sales Analysis

In [1]:

```
import numpy as ap
import pandas as pd
import matplotlib.pyplot as plt #data visualization
import seaborn as sns
```

In [2]:

```
data = pd.read_csv('C:/Users/shrut/Downloads/Python_Diwali_Sales_Analysis/Pyth
```

In [3]:

```
data.head()
```

Out[3]:

| | User_ID | Cust_name | Product_ID | Gender | Age Group | Age | Marital_Status | State | Zc |
|---|---|---|---|---|---|---|---|---|---|
| **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 |

In [4]:

```
data.shape
```

Out[4]:

```
(11251, 15)
```

In [5]:

```
data.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 [6]:

```
#drops unrelated/blank columns
data.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [7]:

```
data.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 [8]:
```
pd.isnull(data)
```

Out[8]:

| | User_ID | Cust_name | Product_ID | Gender | Age Group | Age | Marital_Status | State | Zone | Oc |
|---|---|---|---|---|---|---|---|---|---|---|
| **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(data).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]:
```
data.shape
```

Out[10]:
```
(11251, 13)
```

In [11]:
```
#drop null values
data.dropna(inplace=True)
```

In [12]:
```
data.shape
```

Out[12]:
```
(11239, 13)
```

In [13]:

```
pd.isnull(data).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              0
dtype: int64
```

In [16]:

```
#initialize list of lists
data1=[['Madhav',11],['Shruti',21],['Hari',45],['kehsav', ]]

#creating pandas dataframe using list
df=pd.DataFrame(data1,columns=['Name','Age'])
```

In [17]:

```
df
```

Out[17]:

| | Name | Age |
|---|---|---|
| **0** | Madhav | 11.0 |
| **1** | Shruti | 21.0 |
| **2** | Hari | 45.0 |
| **3** | kehsav | NaN |

In [18]:

```
df.dropna(inplace=True)
#or data2=df.dropna()
```

In [19]:

```
df
```

Out[19]:

| | Name | Age |
|---|---|---|
| **0** | Madhav | 11.0 |
| **1** | Shruti | 21.0 |
| **2** | Hari | 45.0 |

In [21]:

```
#changing datatype
data['Amount']=data['Amount'].astype('int')
```

In [23]:

```
data['Amount'].dtypes
```

Out[23]:

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

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

Out[24]:
```
Index(['Name', 'Age'], dtype='object')
```

In [25]:
```
data.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 [31]:
```
data.rename(columns={'Shadi':'Marital_Status'},inplace=True)
```

In [32]:
```
data
```

Out[32]:

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

11239 rows × 13 columns

In [27]:
```
data.describe()
```

Out[27]:

| | 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 [33]:

```
#using describe for specific column
data[['Age','Orders','Amount']].describe()
```

Out[33]:

| | 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 [34]:

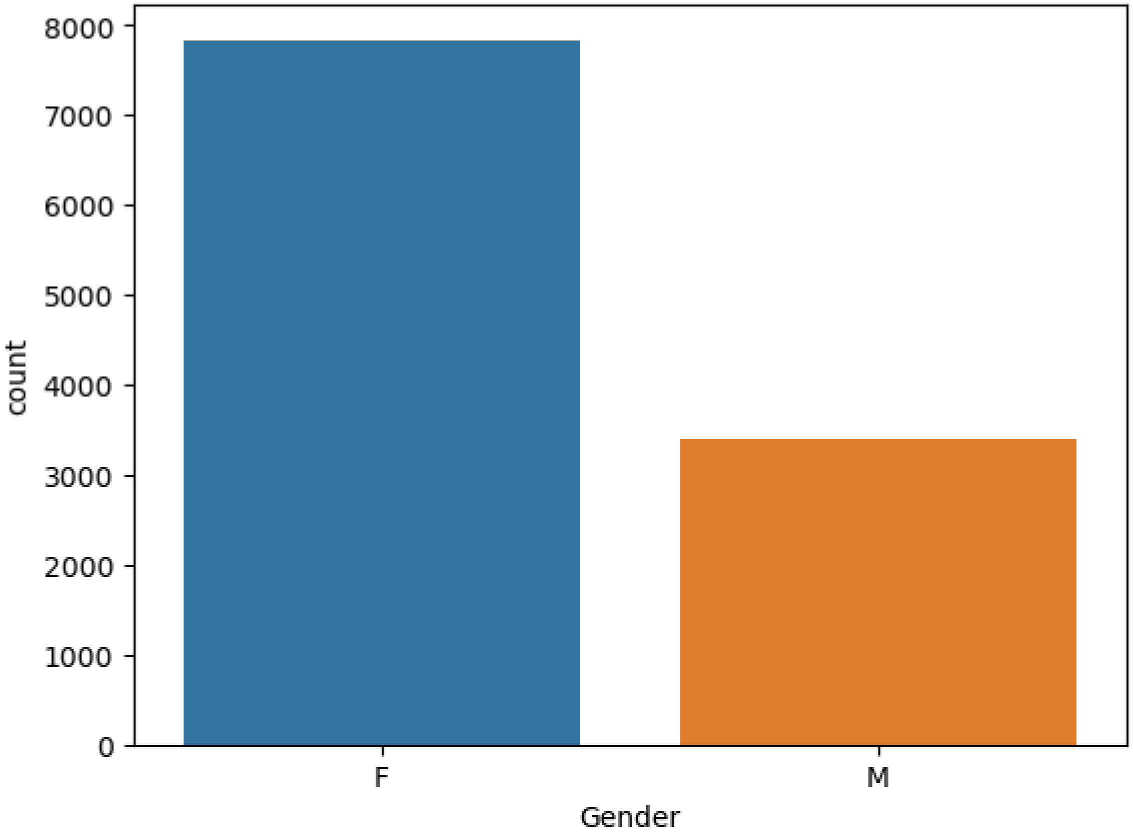
```
data.columns
```

Out[34]:

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

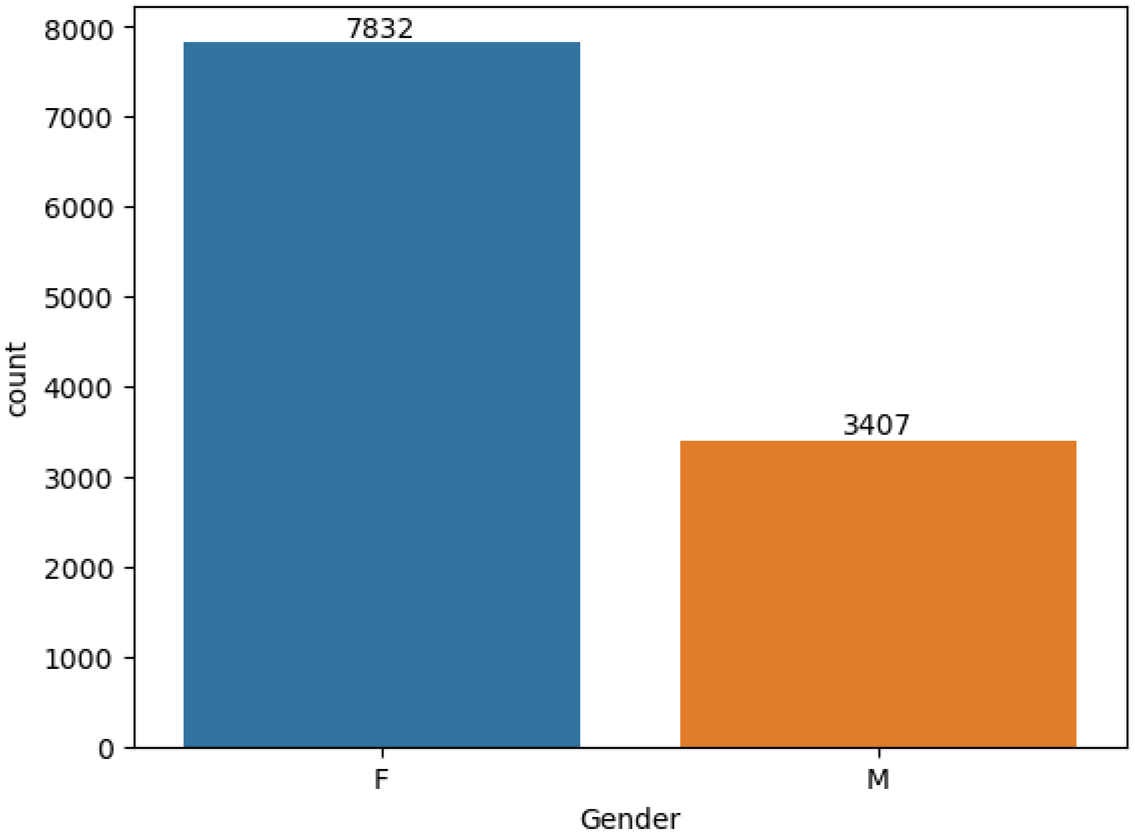
In [35]:
```
sns.countplot(x='Gender',data=data)
```

Out[35]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



In [37]:

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



In [40]:

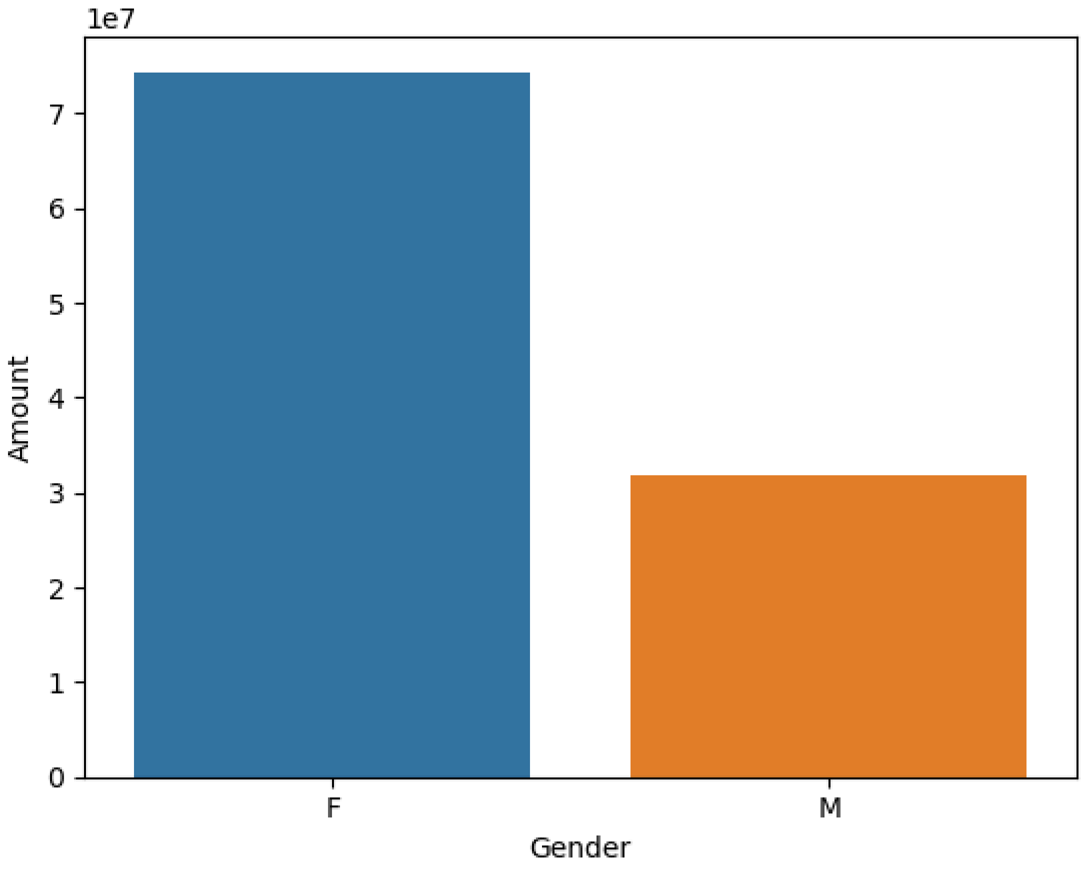
```
data.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount
```

Out[40]:

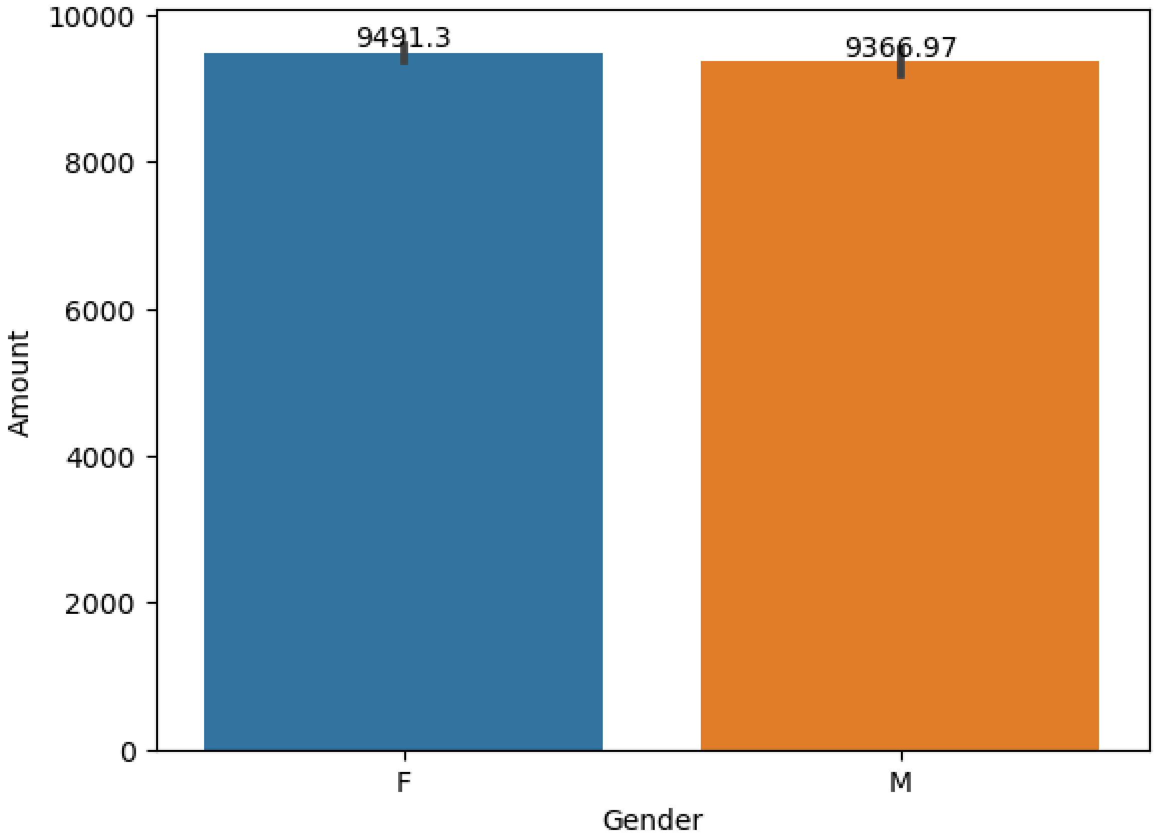
|   | Gender | Amount |
|---|---|---|
| **0** | F | 74335853 |
| **1** | M | 31913276 |

In [55]:
```
sales_gender=data.groupby(['Gender'],as_index=False)['Amount'].sum().sort_valu
sns.barplot(x='Gender',y='Amount',data=sales_gender)
```

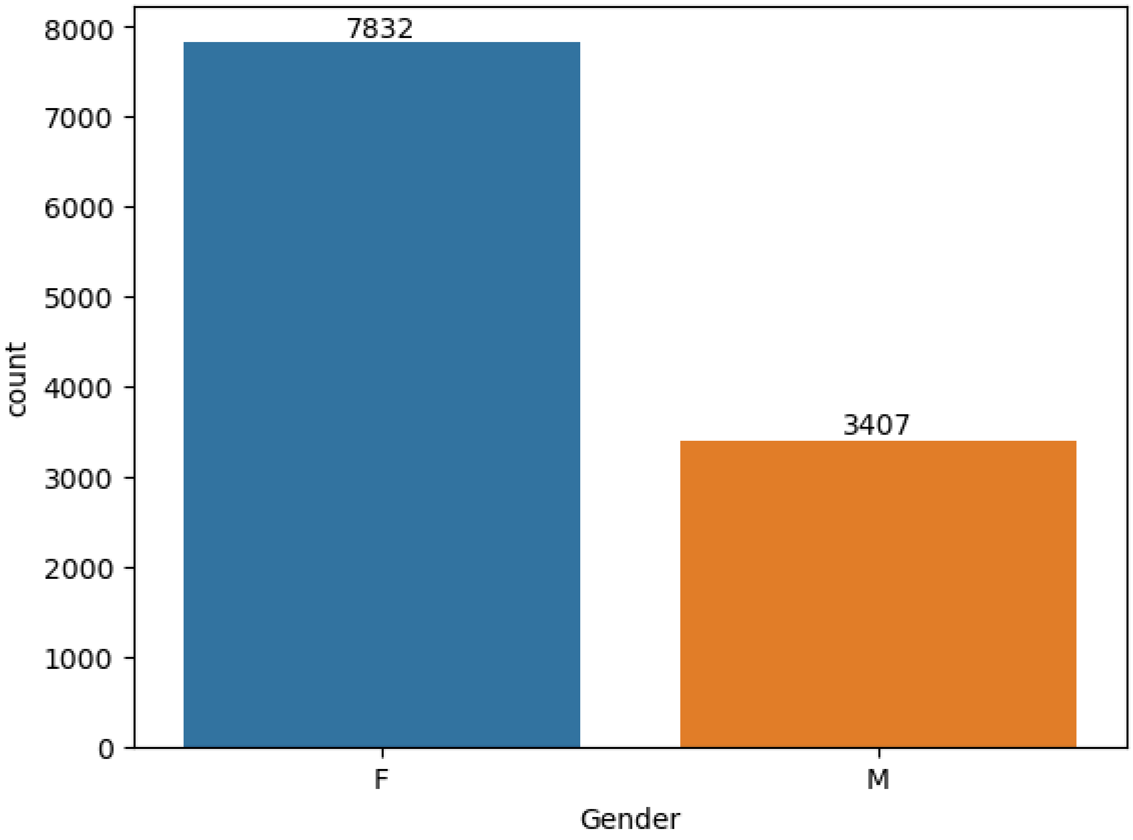
Out[55]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



In [43]:
```
ay=sns.barplot(x='Gender',y='Amount',data=data)
for bars in ay.containers:
    ay.bar_label(bars)
```



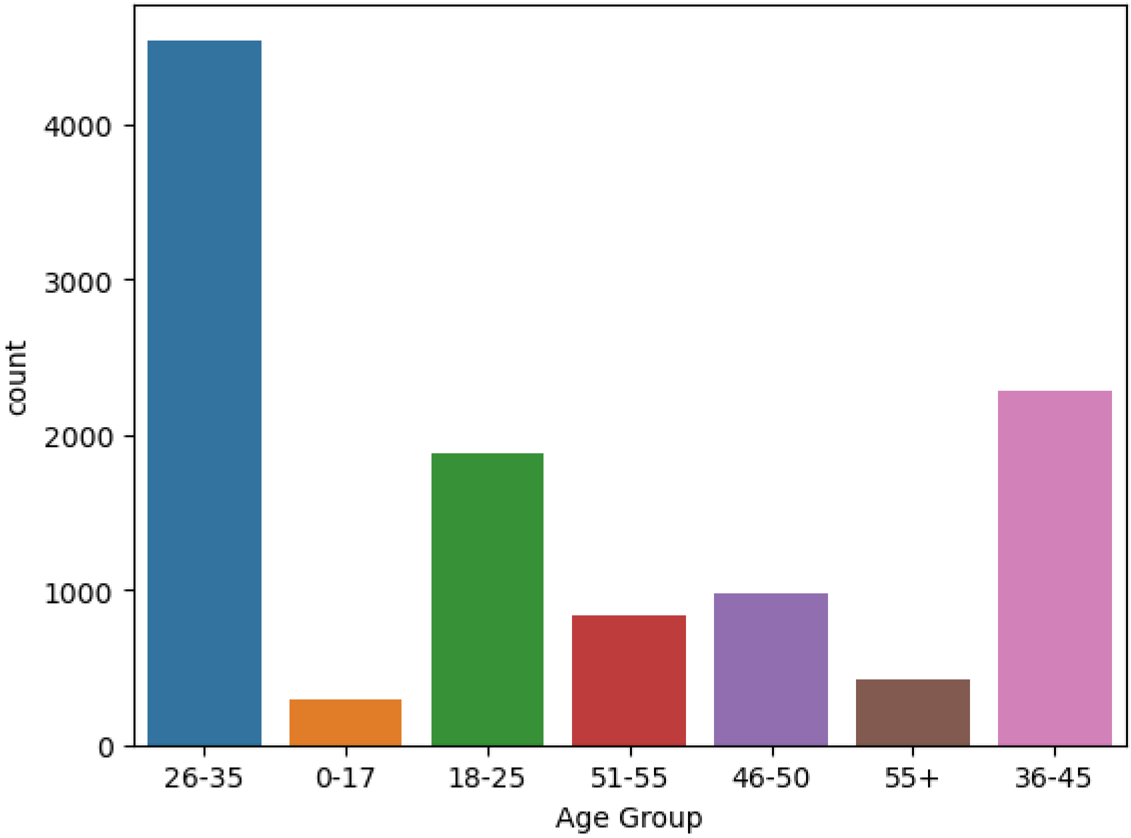
```
In [44]: ay=sns.countplot(x='Gender',data=data)
         for bars in ay.containers:
             ay.bar_label(bars)
```



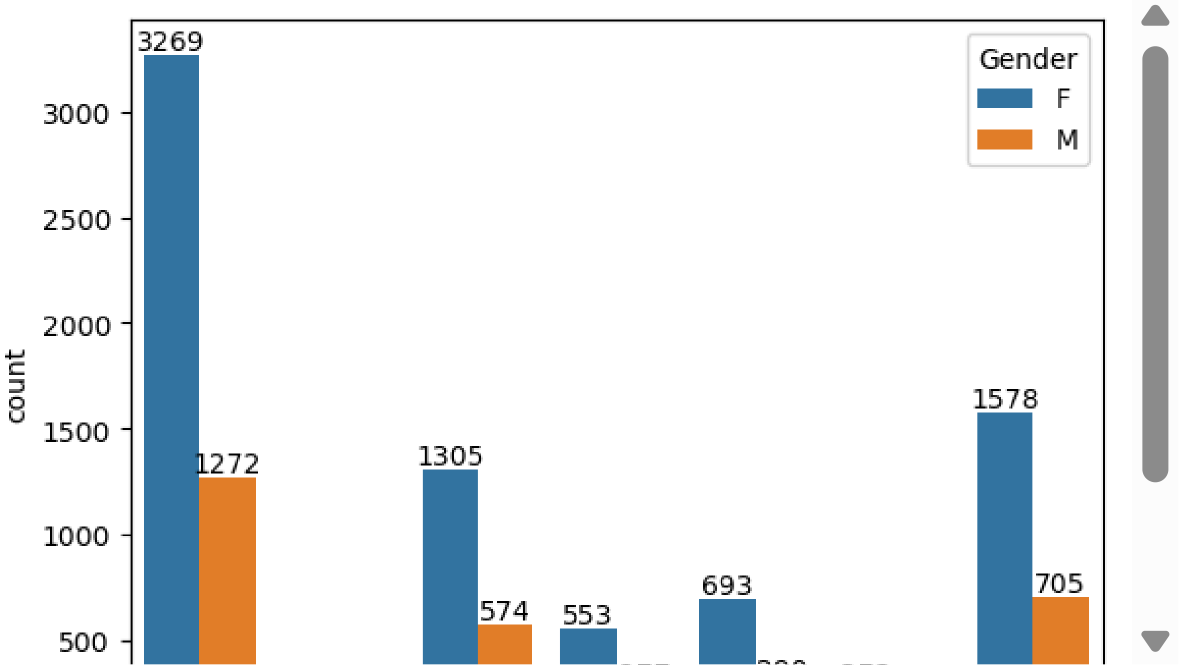
From the above graph we can see that most of the buyers are females and the purchasing power of females are greater than man

AGE

In [45]:
```
az=sns.countplot(x='Age Group',data=data)
for bars in az.containers:
    az.bar_label(bars)
```

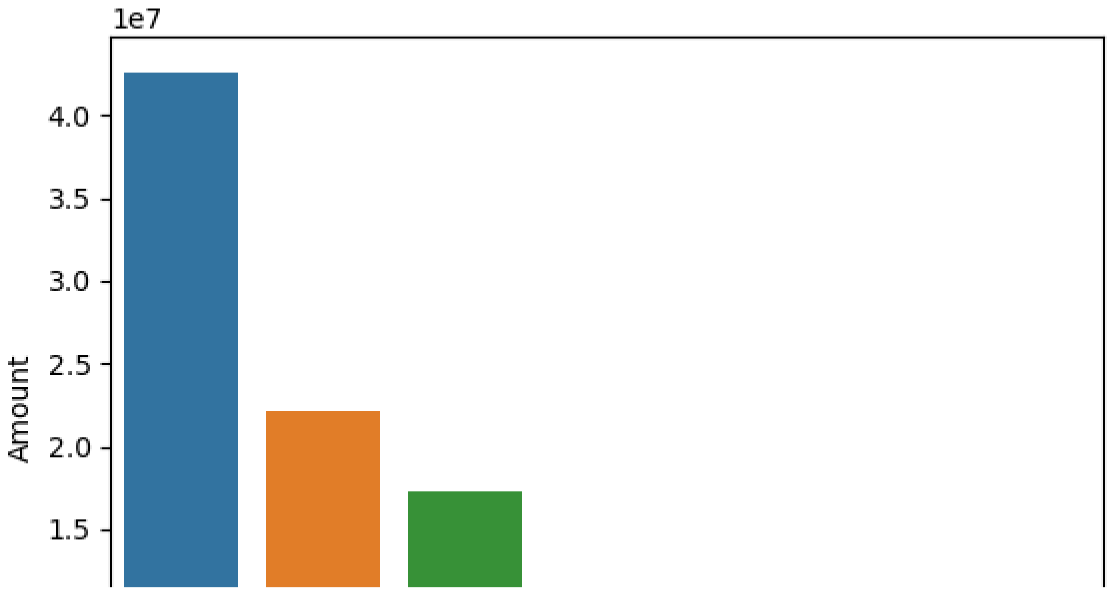


In [53]:
```
az=sns.countplot(x='Age Group',hue='Gender',data=data)
for bars in az.containers:
    az.bar_label(bars)
```



In [54]:
```
sales_age=data.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_valu
sns.barplot(x='Age Group',y='Amount',data=sales_age)
```

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

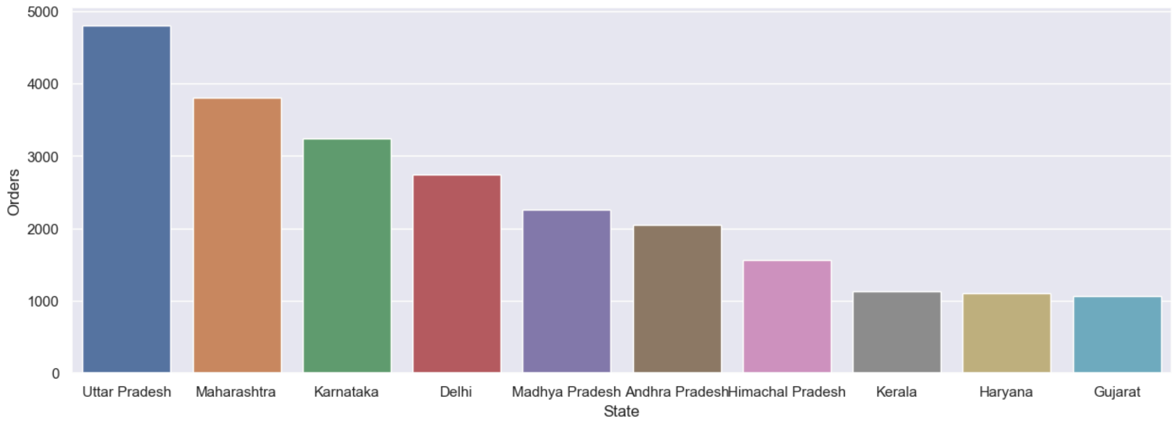


From above graphs we can see that most of the buyers between age group 26-35 years of females

STATE

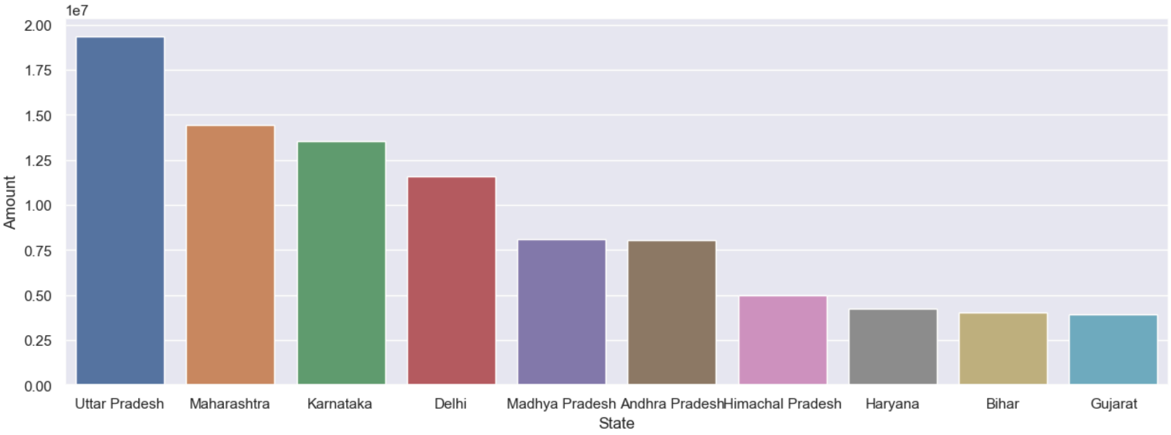
In [62]:
```
#total number of orders from top 10 states
sales_state=data.groupby(['State'],as_index=False)['Orders'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)}) #for plot size
sns.barplot(x='State',y='Orders',data=sales_state)
```

Out[62]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



In [63]:
```
##total amount from top 10 states
sales_state=data.groupby(['State'],as_index=False)['Amount'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)}) #for plot size
sns.barplot(x='State',y='Amount',data=sales_state)
```

Out[63]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



From the above graphs we can see that most of the orders from Uttar Pradesh,Maharastra and Karnataka

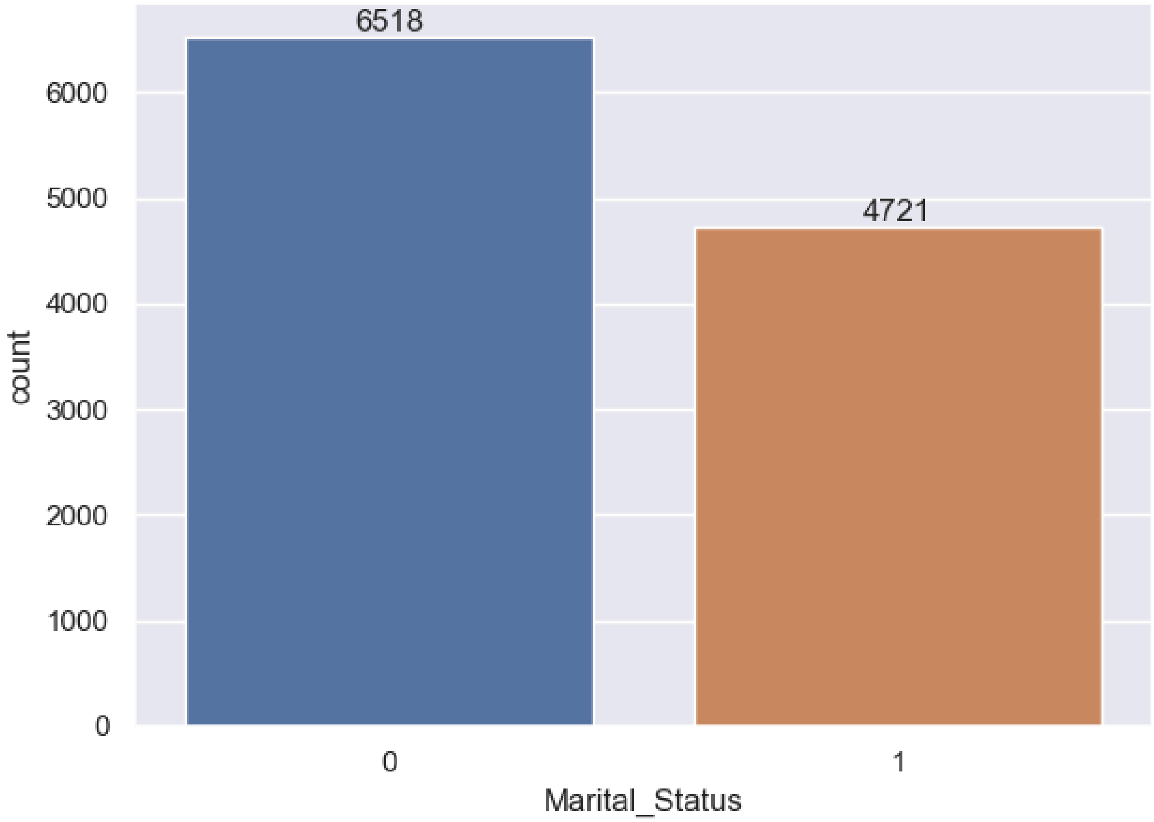
MARITAL STATUS

In [65]:
```
data.columns
```

Out[65]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [69]:

```
az=sns.countplot(x='Marital_Status',data=data)
sns.set(rc={'figure.figsize':(7,3)})
for bars in az.containers:
    az.bar_label(bars)
```

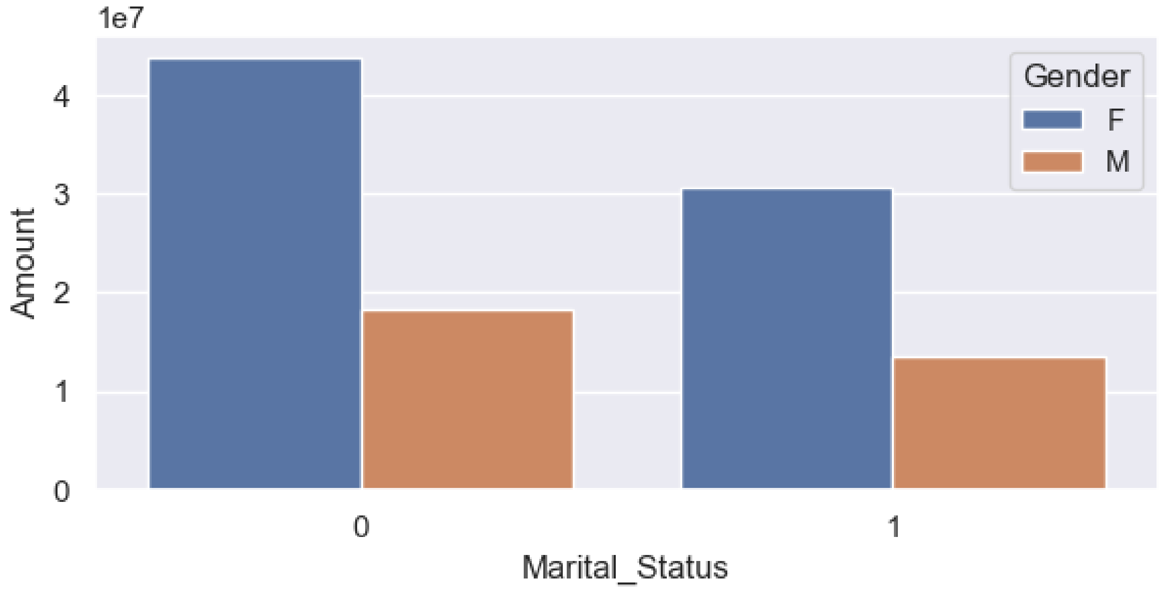


In [74]:

```
sales_marital_state=data.groupby(['Marital_Status','Gender'],as_index=False)['
sns.set(rc={'figure.figsize':(7,3)}) #for plot size
sns.barplot(x='Marital_Status',y='Amount',data=sales_marital_state,hue='Gender
```

Out[74]:

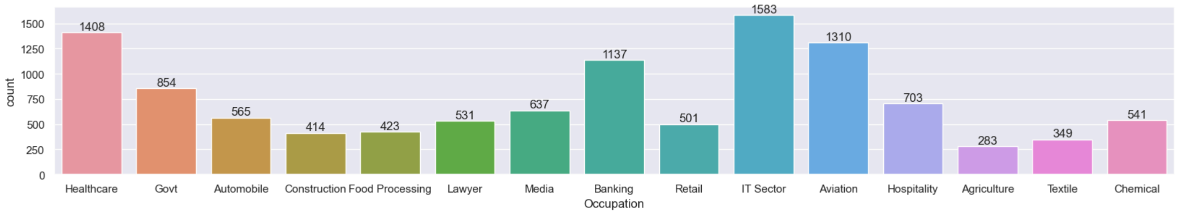
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From the above graph we can see that most of the buyers are married women and their purchasing power is more

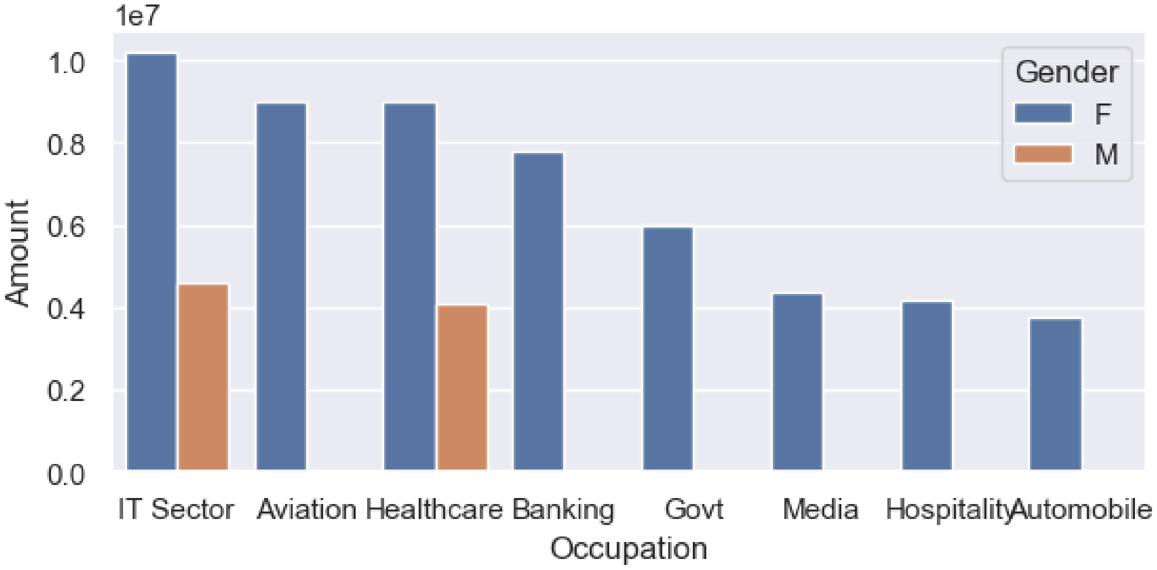
OCCUPATION

In [81]:
```
az=sns.countplot(x='Occupation',data=data)
sns.set(rc={'figure.figsize':(22,3)})
for bars in az.containers:
    az.bar_label(bars)
```



In [83]:
```
sales_occupation=data.groupby(['Occupation','Gender'],as_index=False)['Amount'
sns.set(rc={'figure.figsize':(7,3)}) #for plot size
sns.barplot(x='Occupation',y='Amount',data=sales_occupation,hue='Gender')
```

Out[83]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



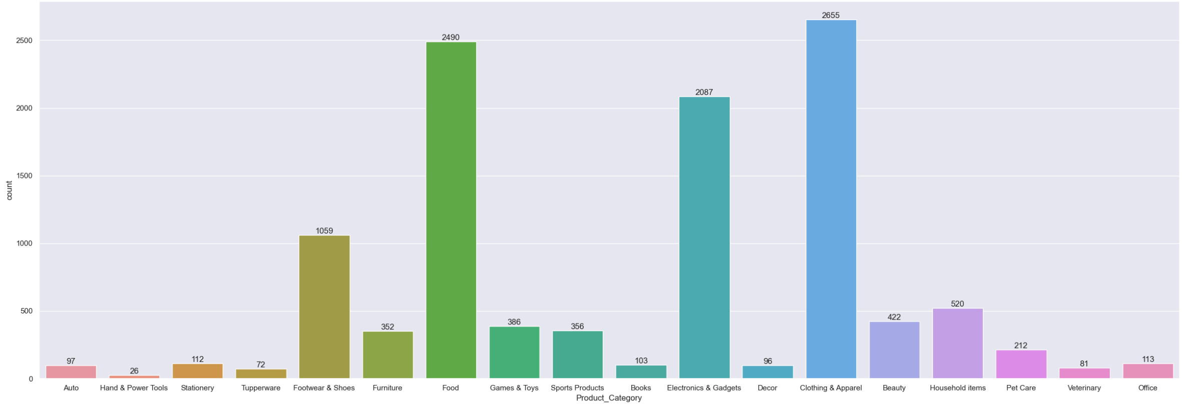
From the above graph it is clear that most of the buyers are working in IT,Healthcare sector and Aviation

PRODUCT CATEGORY

In [84]:
```
data.columns
```

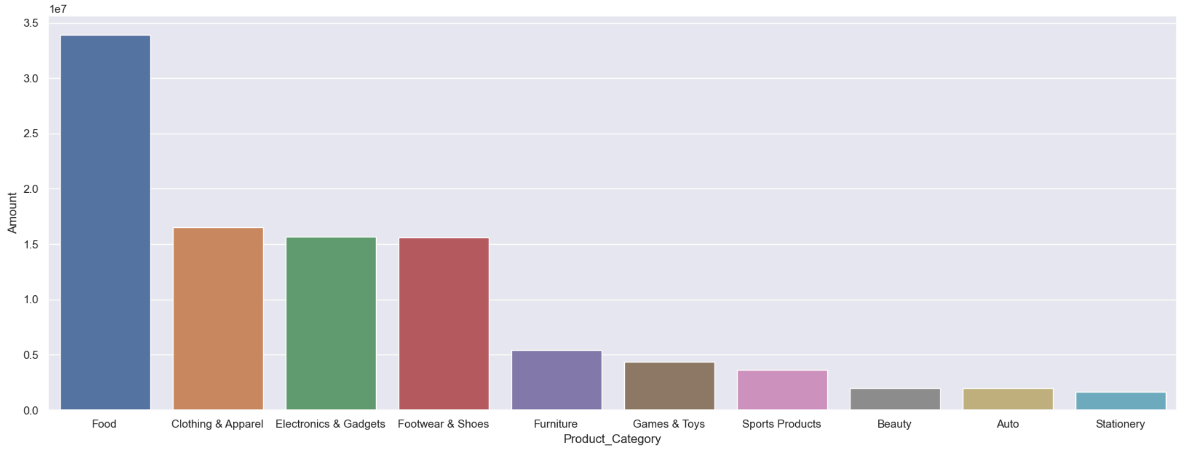
Out[84]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [95]:
```
az=sns.countplot(x='Product_Category',data=data)
sns.set(rc={'figure.figsize':(30,10)})
for bars in az.containers:
    az.bar_label(bars)
```



In [91]:
```
sales_product=data.groupby(['Product_Category'],as_index=False)['Amount'].sum(
sns.set(rc={'figure.figsize':(20,7)}) #for plot size
sns.barplot(x='Product_Category',y='Amount',data=sales_product)
```

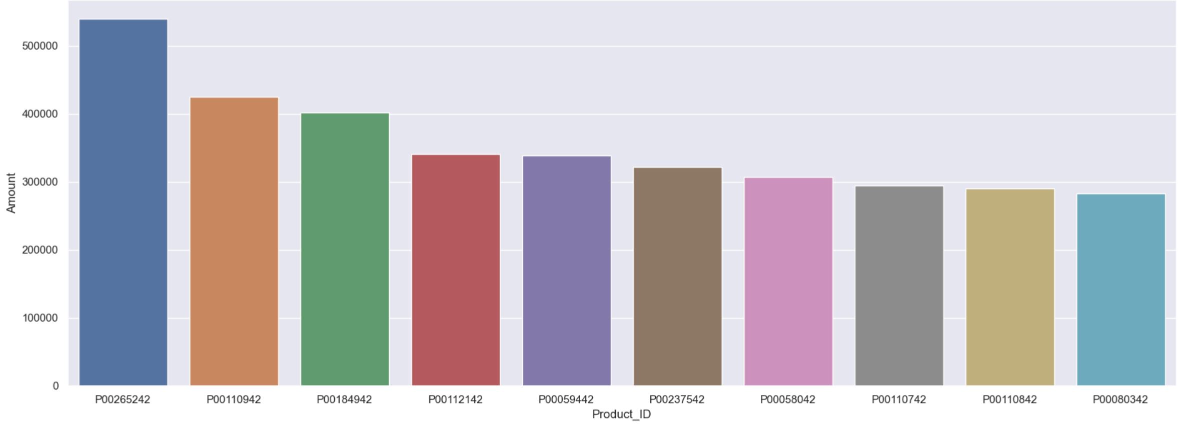
Out[91]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



From the above graph we can see that most of the sold products are from Food,Clothing and Electronics

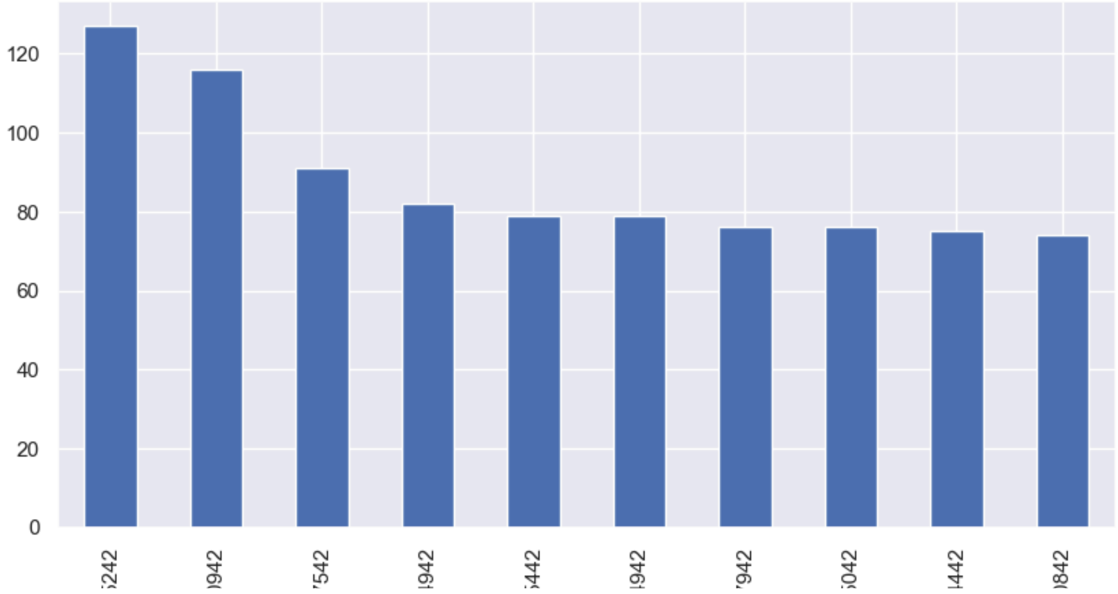
In [97]:
```
sales_product=data.groupby(['Product_ID'],as_index=False)['Amount'].sum().sort
sns.set(rc={'figure.figsize':(20,7)}) #for plot size
sns.barplot(x='Product_ID',y='Amount',data=sales_product)
```

Out[97]:
```
<Axes: xlabel='Product_ID', ylabel='Amount'>
```



In [100]:
```
#top 10 most sold products(same as above)
fig1,az1=plt.subplots(figsize=(10,5))
data.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=
```

Out[100]:
```
<Axes: xlabel='Product_ID'>
```



# Conclusion

Married women age group 26-35 years from UP,Maharastra and karnataka working in IT,Healthcare Sectors and aviation are likely to buy more products from Food,Clothing and Electronics Category