In [1]:
```
Diwali Sales Analysis
```

In [ ]:
```
Objective :
    >> Improve customer experience by analyzing data
    >> Increase revenue
```

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [16]:
```
df = pd.read_csv("Diwali Sales Data 1.csv")
df
```

Out[16]:

	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	Sou
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	C
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sou
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W
...	...	...	...	...	...	...	...	...	
5472	1000889	Parth	P00248942	F	46-50	47	0	Punjab	No
5473	1000793	Staavos	P00288042	M	36-45	42	1	Bihar	E
5474	1002793	Aniket	P00288742	F	36-45	39	1	Madhya Pradesh	C
5475	1002890	Maithilee	P00037142	F	36-45	41	0	Rajasthan	No
5476	1000850	Bhawna	P00105442	F	36-45	42	1	Uttar Pradesh	C

5477 rows × 13 columns

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

Out[17]:
```
(5477, 13)
```

In [18]:
```
df.head()
```

Out[18]:

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

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5477 entries, 0 to 5476
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           5477 non-null   int64  
 1   Cust_name         5477 non-null   object 
 2   Product_ID        5477 non-null   object 
 3   Gender            5477 non-null   object 
 4   Age Group         5477 non-null   object 
 5   Age               5477 non-null   int64  
 6   Marital_Status    5477 non-null   int64  
 7   State             5477 non-null   object 
 8   Zone              5477 non-null   object 
 9   Occupation        5477 non-null   object 
 10  Product_Category  5477 non-null   object 
 11  Orders            5477 non-null   int64  
 12  Amount            5470 non-null   float64
dtypes: float64(1), int64(4), object(8)
memory usage: 556.4+ KB
```

In [20]:
```
#Checking for null values
pd.isnull(df).sum()
```

Out[20]:
```
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             7
dtype: int64
```

In [21]:
```
#drop null values
df.dropna(inplace = True)
```

In [22]:
```
df.shape
```

Out[22]:
```
(5470, 13)
```

In [23]:
```
#change data types
df['Amount'] = df['Amount'].astype('int')
```

In [24]:
```
df['Amount'].dtypes
```

Out[24]:
```
dtype('int64')
```

In [25]:
```
df.describe()
```

Out[25]:

	User_ID	Age	Marital_Status	Orders	Amount
count	5.470000e+03	5470.000000	5470.000000	5470.000000	5470.000000
mean	1.002977e+06	35.745521	0.411883	2.475868	13306.035832
std	1.712806e+03	12.828031	0.492219	1.111478	3898.407908
min	1.000003e+06	12.000000	0.000000	1.000000	7953.000000
25%	1.001465e+06	27.000000	0.000000	1.000000	9916.250000
50%	1.003041e+06	33.000000	0.000000	2.000000	12427.000000
75%	1.004406e+06	44.000000	1.000000	3.000000	16148.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

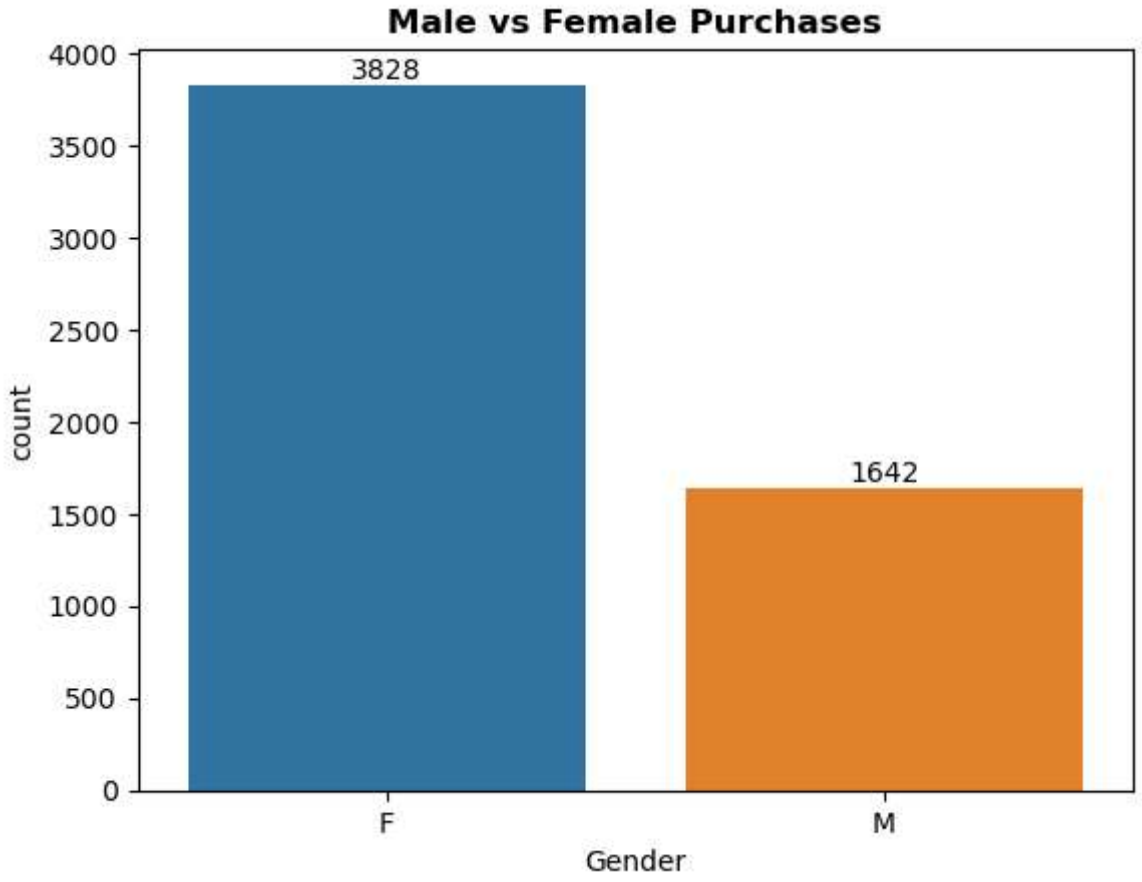
In [26]:
```
#describe for specific columns

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

Out[26]:

	Amount	Age	Orders
count	5470.000000	5470.000000	5470.000000
mean	13306.035832	35.745521	2.475868
std	3898.407908	12.828031	1.111478
min	7953.000000	12.000000	1.000000
25%	9916.250000	27.000000	1.000000
50%	12427.000000	33.000000	2.000000
75%	16148.000000	44.000000	3.000000
max	23952.000000	92.000000	4.000000

In [27]:
```
fig = sns.countplot(x = 'Gender', data = df)
```

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

plt.title("Male vs Female Purchases", fontweight = 'bold')
plt.show()
```

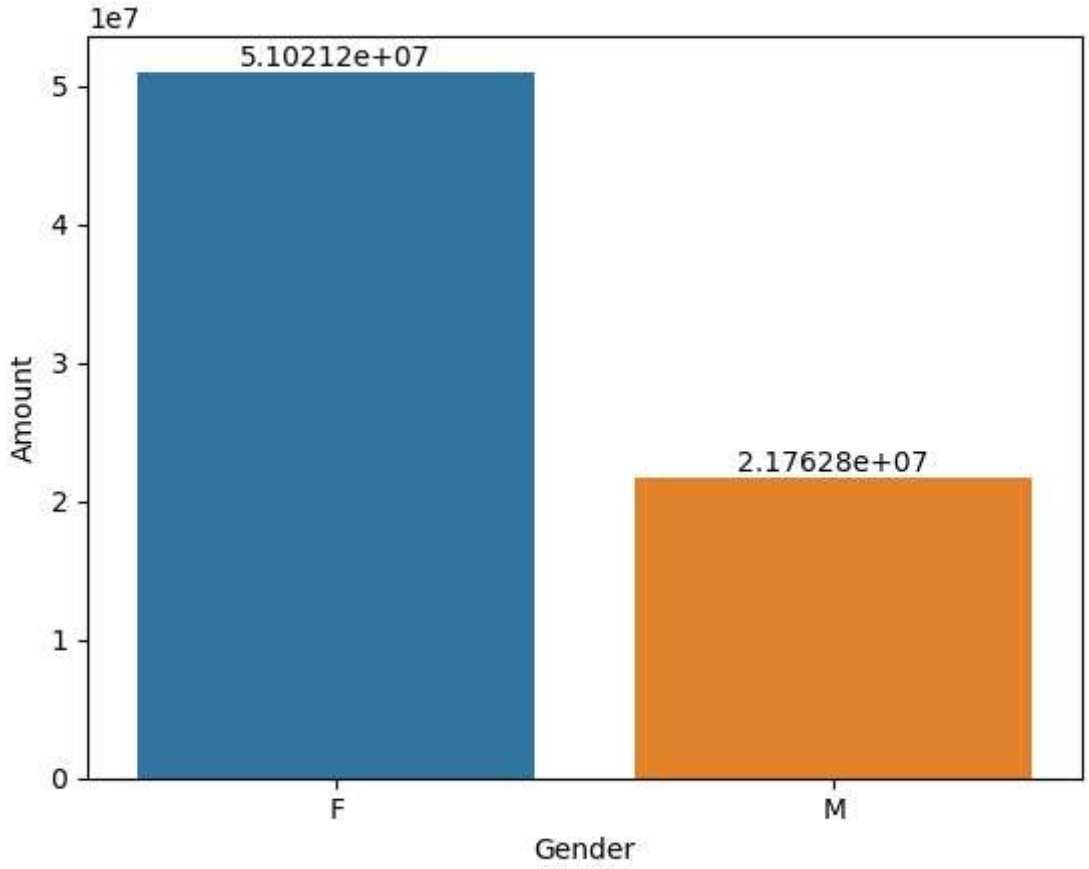


In [28]:
```
gender = df.groupby(['Gender'], as_index = False)['Amount'].sum().sort_values(by='A
gender
```

Out[28]:

	Gender	Amount
0	F	51021240
1	M	21762776

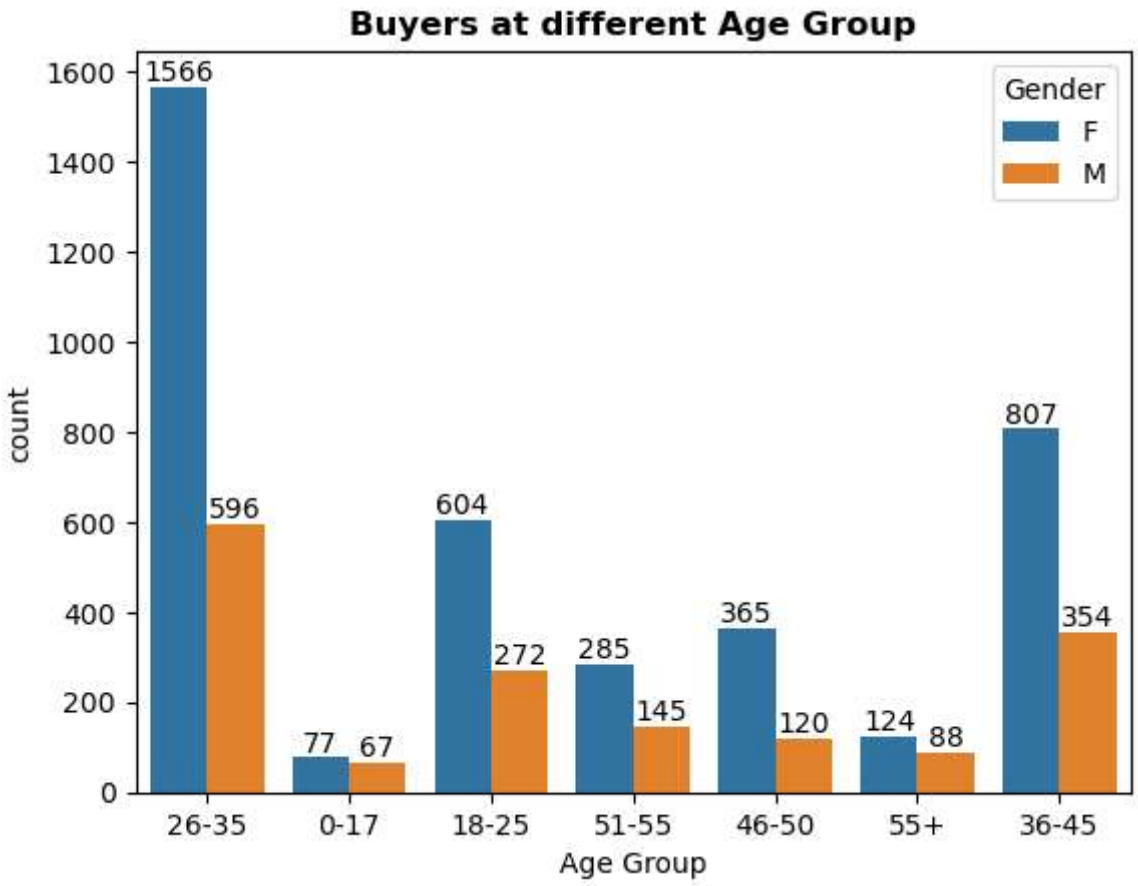
In [29]:
```
fig = sns.barplot(x = 'Gender', y = 'Amount', data = gender)

for bars in fig.containers:
    fig.bar_label(bars)

plt.show()
```



In [ ]:
```
#From above graph we find that most of the buyers are female and purchasing power o
```

In [ ]:
```
```

In [30]:
```
fig = sns.countplot(x = 'Age Group', data = df, hue = 'Gender')

for bars in fig.containers:
    fig.bar_label(bars)


plt.title('Buyers at different Age Group', fontweight = 'bold')
plt.show()
```

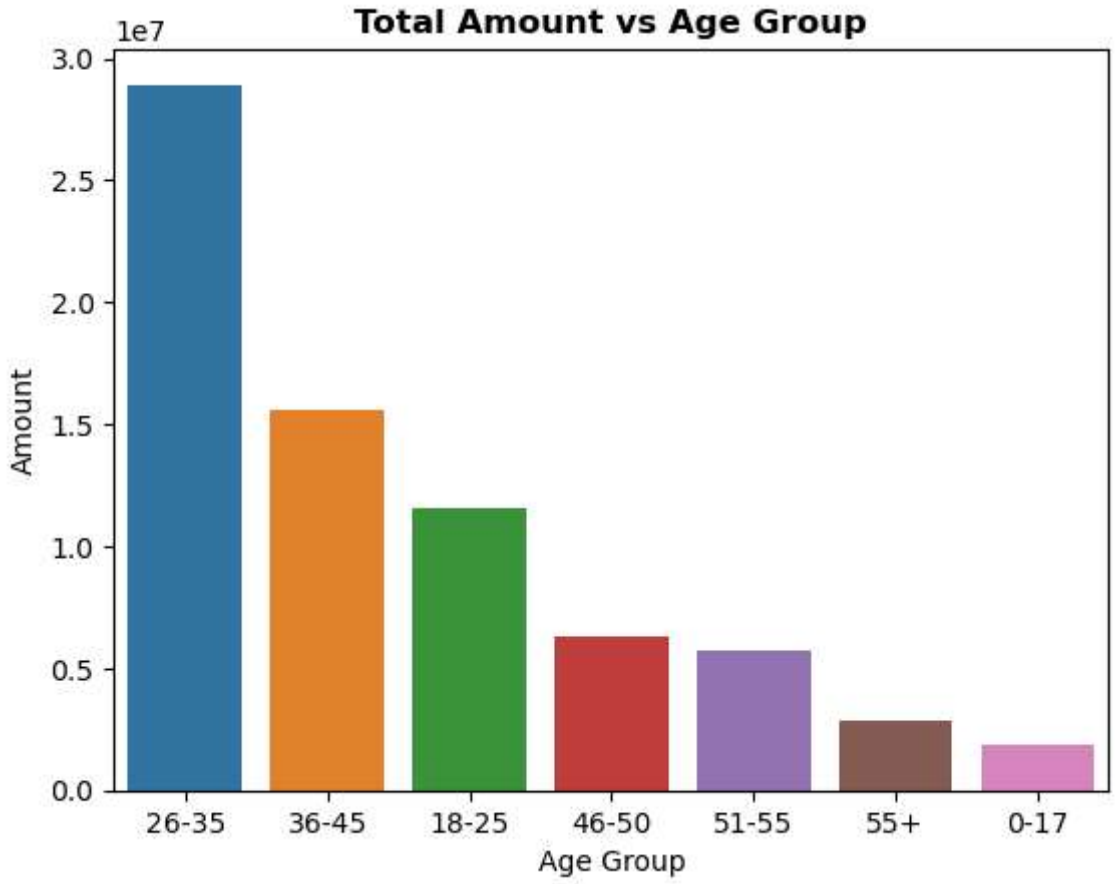


In [31]:
```
sales_age = df.groupby(['Age Group'], as_index = False)['Amount'].sum().sort_values
sales_age
```

Out[31]:

	Age Group	Amount
2	26-35	28915566
3	36-45	15604036
1	18-25	11529802
4	46-50	6315524
5	51-55	5754442
6	55+	2815055
0	0-17	1849591

In [32]:
```
sns.barplot(x = 'Age Group', y = 'Amount', data = sales_age)

plt.title('Total Amount vs Age Group', fontweight = 'bold')

plt.show()
```



In [ ]:
```
#From above graph we can find that most of the buyers are of age group 26-35
```

In [ ]:
```

```

In [ ]:
```
#Total Number of orders from Top 10 states
```

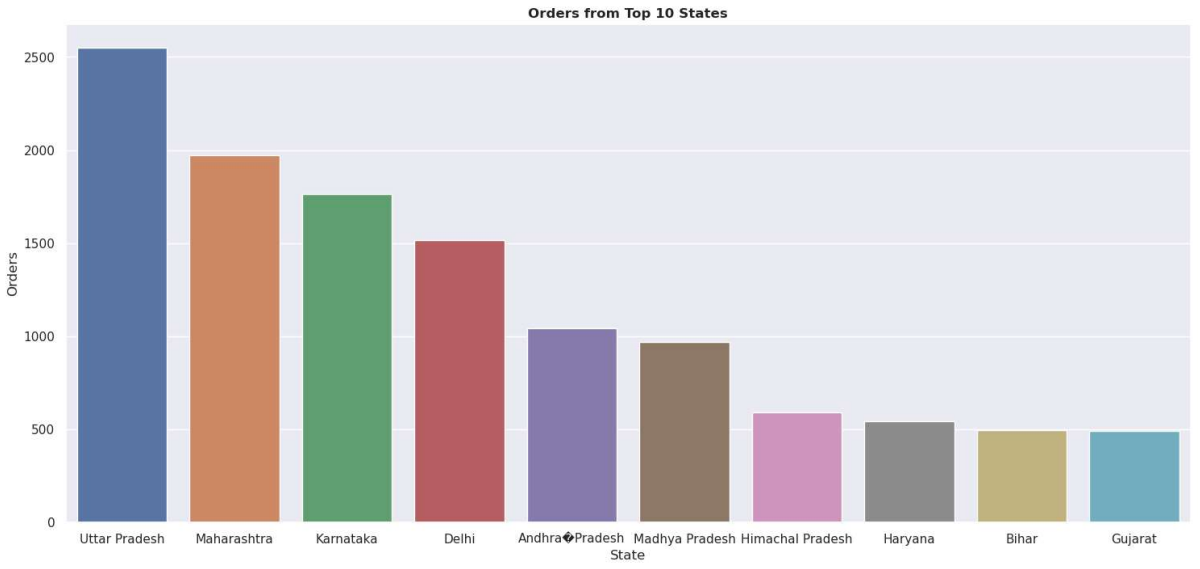
In [33]:
```
sale_state = df.groupby(['State'], as_index = False)['Orders'].sum().sort_values(by
sale_state
```

Out[33]:

	State	Orders
14	Uttar Pradesh	2548
10	Maharashtra	1971
7	Karnataka	1764
2	Delhi	1518
0	Andhra🔲Pradesh	1044
9	Madhya Pradesh	969
5	Himachal Pradesh	592
4	Haryana	542
1	Bihar	494
3	Gujarat	492

In [34]:
```
sns.set(rc={'figure.figsize':(18,8)})
```

In [35]:
```
sns.barplot(x = 'State', y = 'Orders' , data = sale_state)
plt.title('Orders from Top 10 States', fontweight = 'bold')
plt.show()
```



In [ ]:
```
#From above we can say that maximum orders are recieved from Uttar Pradesh, Maharas
```

In [ ]:
```

```

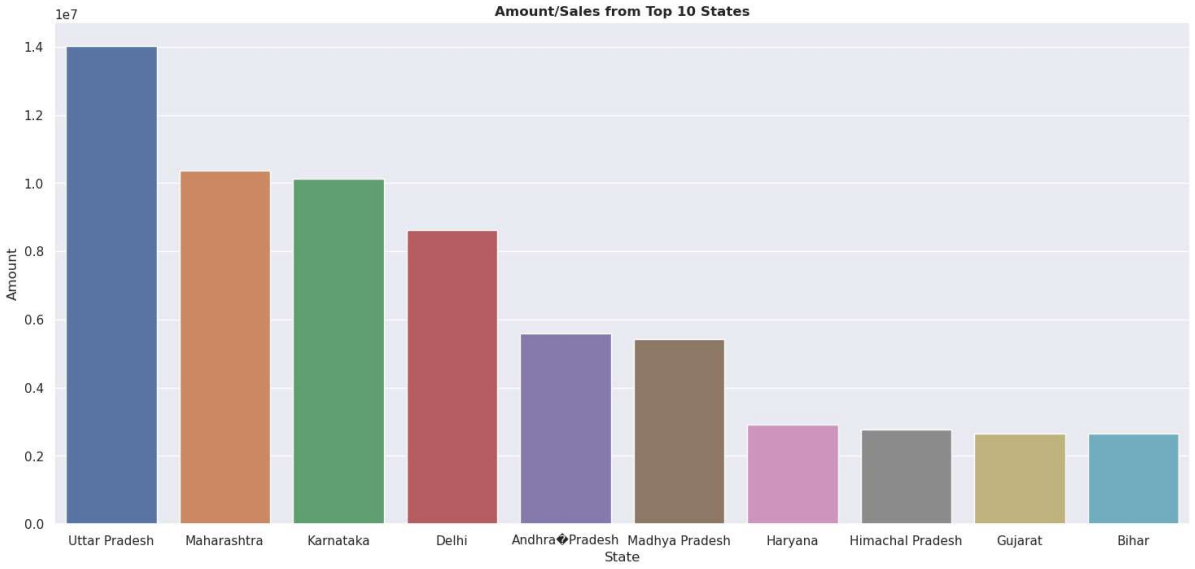
In [ ]:
```
#Total Amount/Sales from Top 10 states
```

In [36]:
```
sale_state = df.groupby(['State'], as_index = False)['Amount'].sum().sort_values(by
sale_state
```

Out[36]:

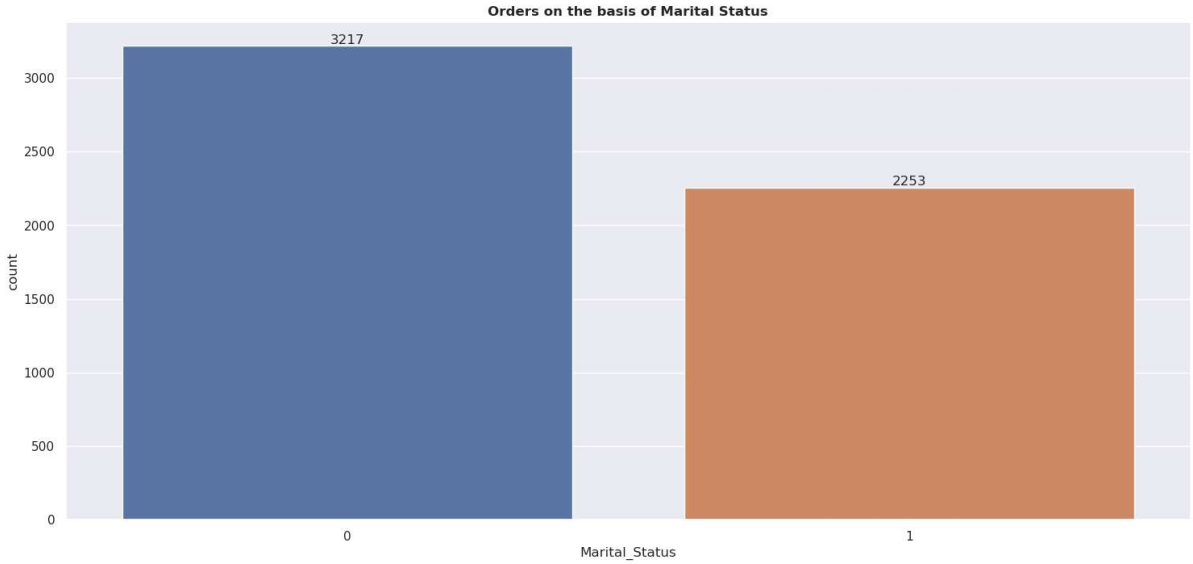
	State	Amount
14	Uttar Pradesh	14010018
10	Maharashtra	10369699
7	Karnataka	10121254
2	Delhi	8612932
0	Andhra Pradesh	5591619
9	Madhya Pradesh	5419181
4	Haryana	2913215
5	Himachal Pradesh	2770862
3	Gujarat	2662720
1	Bihar	2650061

In [37]:
```
sns.set(rc={'figure.figsize':(18,8)})
sns.barplot(x = 'State', y = 'Amount' , data = sale_state)
plt.title('Amount/Sales from Top 10 States', fontweight = 'bold')
plt.show()
```



In [ ]:
```
```

In [38]:
```
fig = sns.countplot(x = 'Marital_Status', data = df)

for bars in fig.containers:
    fig.bar_label(bars)

sns.set(rc={'figure.figsize':(10,6)})
plt.title('Orders on the basis of Marital Status', fontweight = 'bold', fontsize =
plt.show()
```

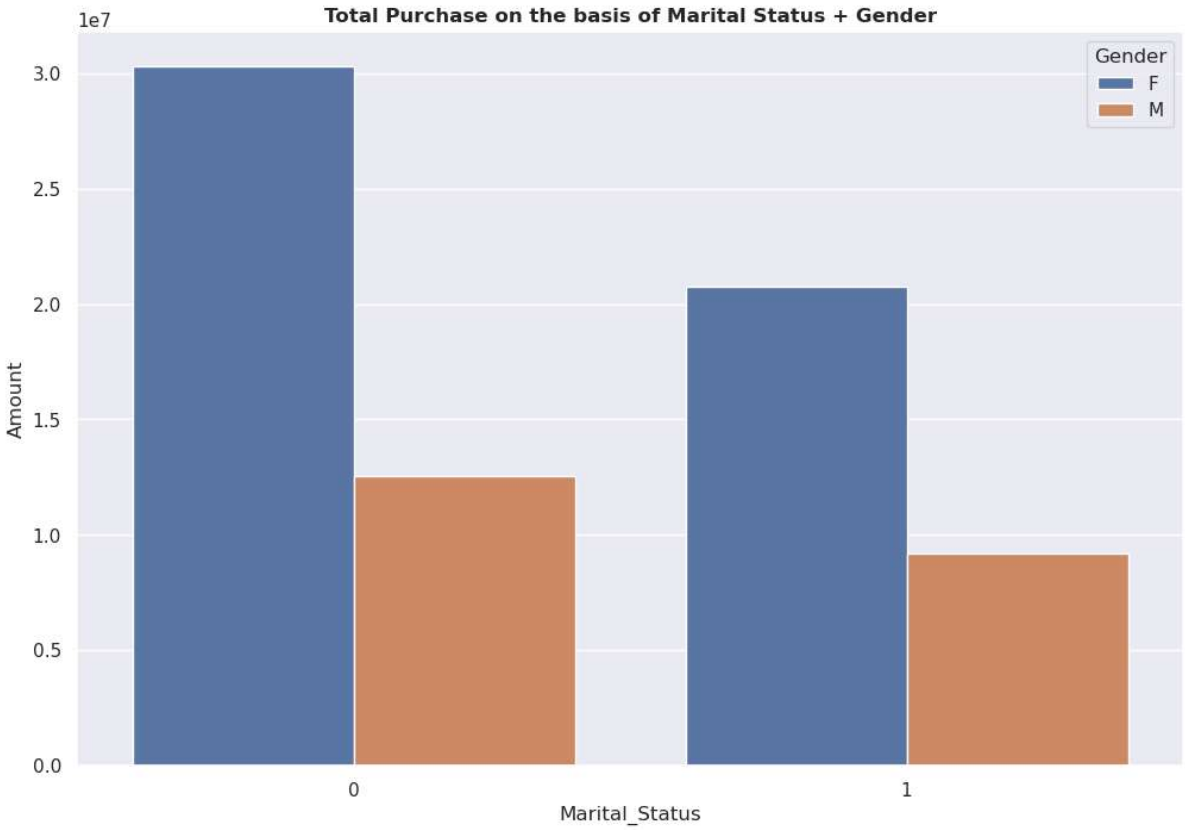


In [39]:
```
marital_status = df.groupby(['Marital_Status', 'Gender'], as_index = False)['Amount
marital_status
```

Out[39]:

	Marital_Status	Gender	Amount
0	0	F	30280975
2	1	F	20740265
1	0	M	12554346
3	1	M	9208430

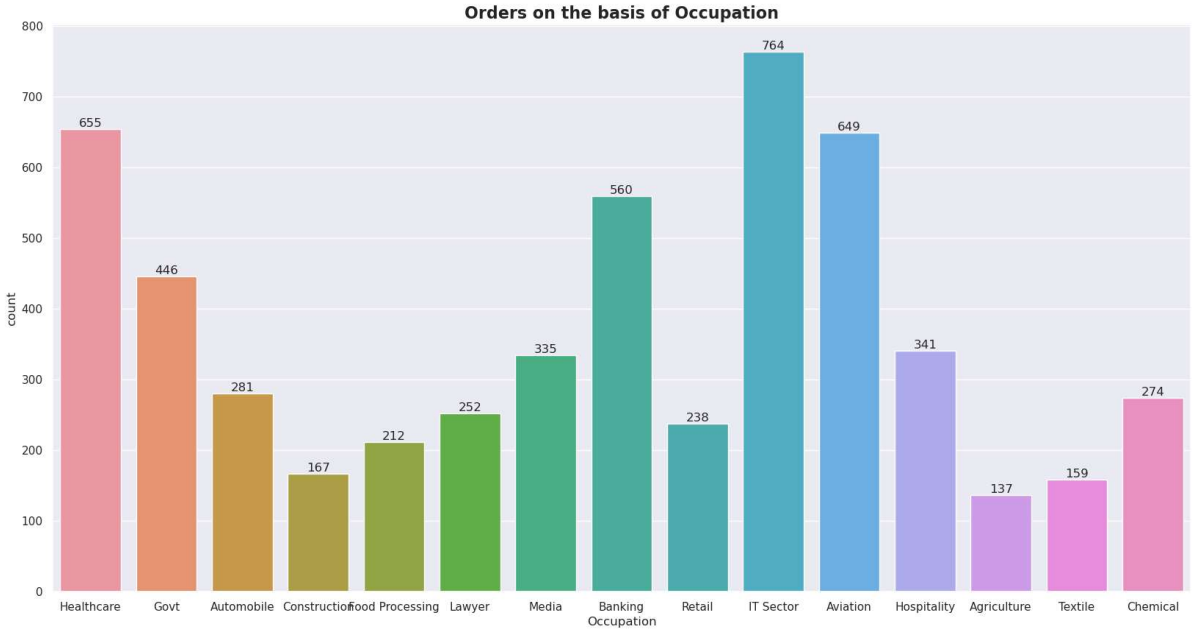
In [40]:
```
sns.set(rc={'figure.figsize':(12,8)})
sns.barplot(x = 'Marital_Status', y = 'Amount', hue = 'Gender' , data = marital_sta
plt.title('Total Purchase on the basis of Marital Status + Gender', fontweight = 'b
plt.show()
```



In [ ]:
```
#From above graph we can say that most of the buyers are married(Female) and they h
```

In [ ]:
```

```

In [41]:
```
sns.set(rc={'figure.figsize' : (20,10)})
fig = sns.countplot(data = df, x = 'Occupation')

plt.title('Orders on the basis of Occupation', fontweight = 'bold', fontsize = 16)


for bars in fig.containers:
    fig.bar_label(bars)


plt.show()
```



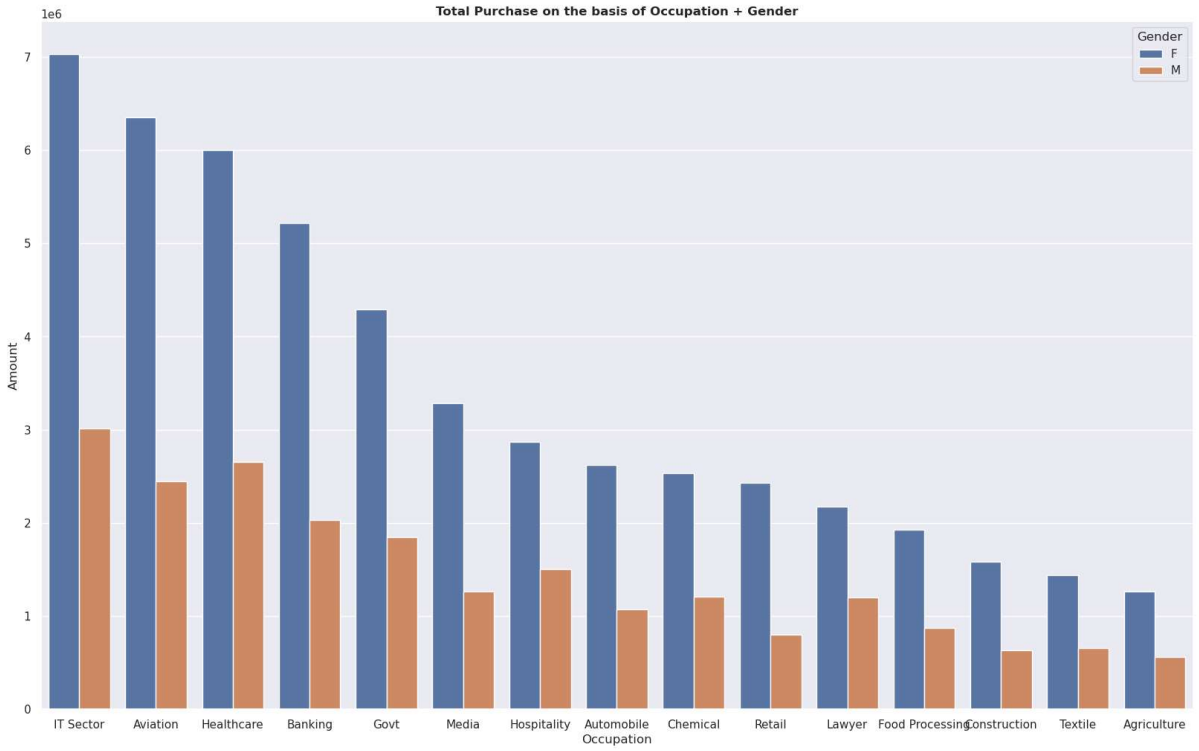
In [42]:
```
occupation = df.groupby(['Occupation', 'Gender'], as_index = False)['Amount'].sum()
occupation
```

Out[42]:

	Occupation	Gender	Amount
20	IT Sector	F	7028862
4	Aviation	F	6355016
16	Healthcare	F	6000192
6	Banking	F	5220466
14	Govt	F	4294650
24	Media	F	3286001
21	IT Sector	M	3015915
18	Hospitality	F	2871436
17	Healthcare	M	2650683
2	Automobile	F	2625100
8	Chemical	F	2532954
5	Aviation	M	2448396
26	Retail	F	2432887
22	Lawyer	F	2170705
7	Banking	M	2030748
12	Food Processing	F	1923964
15	Govt	M	1846772
10	Construction	F	1580530
19	Hospitality	M	1505572
28	Textile	F	1438025
25	Media	M	1264054
0	Agriculture	F	1260452
9	Chemical	M	1204269
23	Lawyer	M	1201463
3	Automobile	M	1069533
13	Food Processing	M	873032
27	Retail	M	802717
29	Textile	M	660813
11	Construction	M	630505
1	Agriculture	M	558304

In [43]:
```
sns.set(rc={'figure.figsize':(20,12)})
sns.barplot(x = 'Occupation', y = 'Amount', hue = 'Gender' , data = occupation)
```

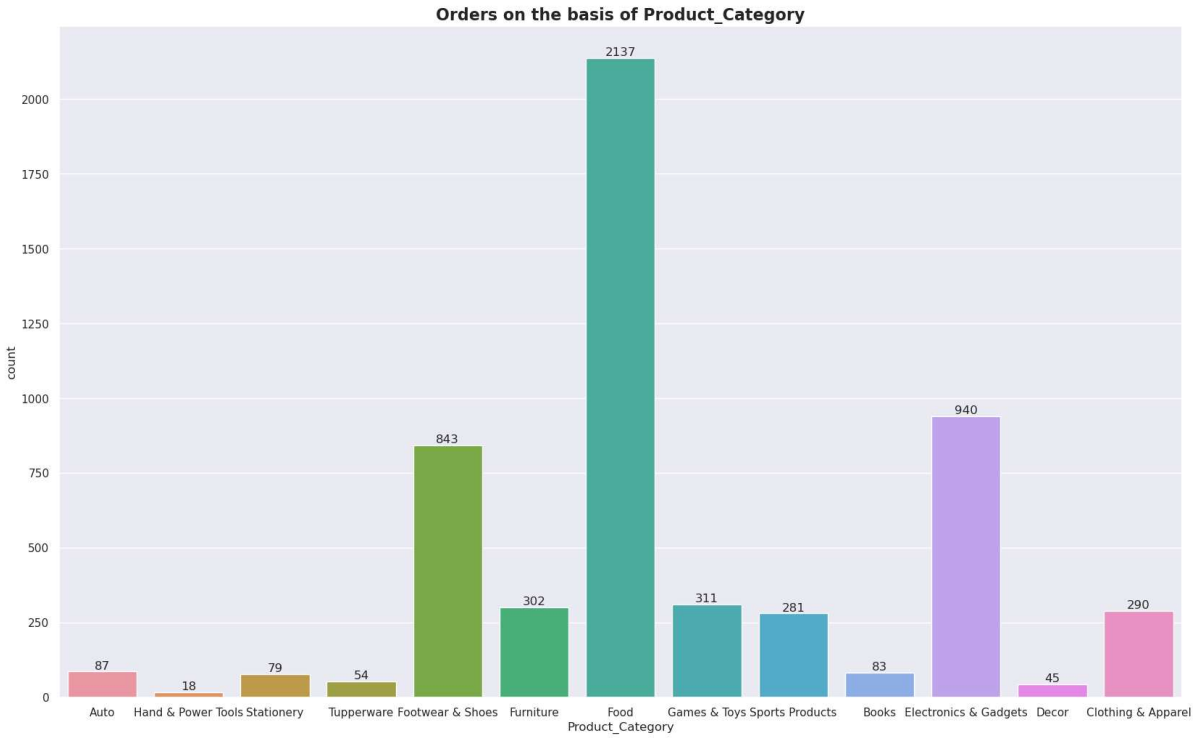
```
plt.title('Total Purchase on the basis of Occupation + Gender', fontweight = 'bold'
plt.show()
```



In [ ]:
```
#From above graph we can say that most of the buyers are from IT Sector, Aviation,
```

In [ ]:
```

```

In [44]:
```
fig = sns.countplot(x = 'Product_Category', data = df)

for bars in fig.containers:
    fig.bar_label(bars)

plt.title('Orders on the basis of Product_Category', fontweight = 'bold', fontsize
plt.show()
```

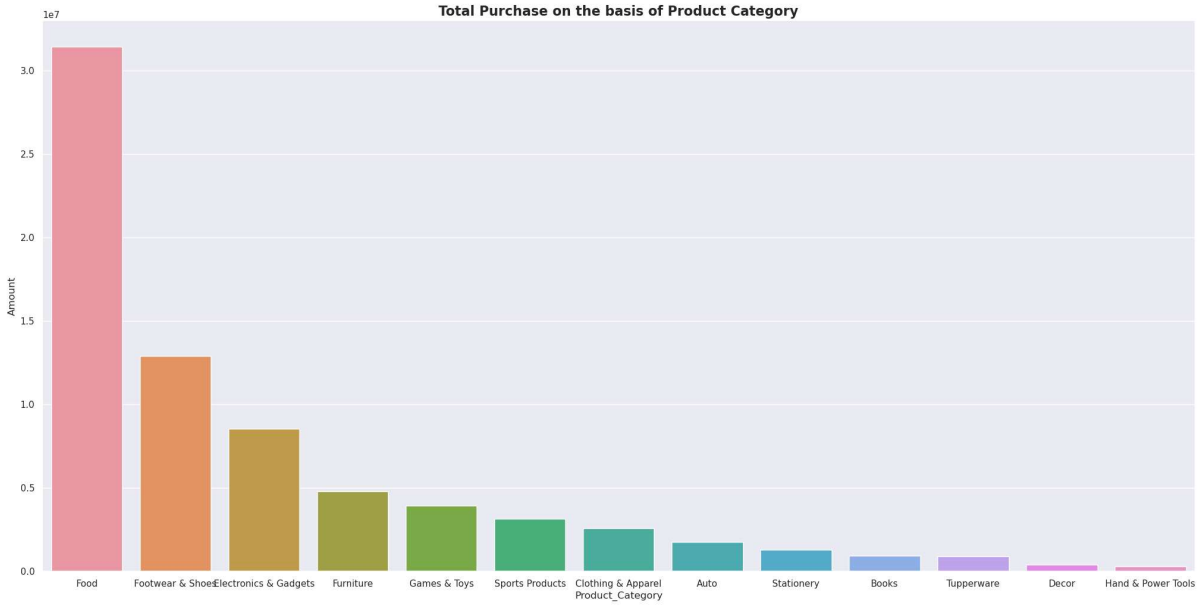


In [45]:
```
product = df.groupby(['Product_Category'], as_index = False)['Amount'].sum().sort_v
product
```

Out[45]:

	Product_Category	Amount
5	Food	31418952
6	Footwear & Shoes	12892366
4	Electronics & Gadgets	8541168
7	Furniture	4772467
8	Games & Toys	3913943
10	Sports Products	3147294
2	Clothing & Apparel	2558486
0	Auto	1744828
11	Stationery	1281655
1	Books	919827
12	Tupperware	889160
3	Decor	411953
9	Hand & Power Tools	291917

In [49]:
```
sns.set(rc={'figure.figsize':(25,12)})
sns.barplot(x = 'Product_Category', y = 'Amount' , data = product)
plt.title('Total Purchase on the basis of Product Category', fontweight = 'bold' ,
plt.show()
```



In [ ]:

In [ ]:
```
#Conclusion :

Married Women of age_group 26-35 yrs from Uttar Pradesh, Maharashtra and Karnataka
working in IT Sector, Aviation and Healthcare are more likely to buy products such
```

In [ ]: