In [2]:

```
import pandas as pd
```

In [3]:

```
import numpy as np
```

In [4]:

```
import seaborn as sns
```

In [5]:

```
df=pd.read_csv("C:\\Users\\USER\\Downloads\\supermarket.csv")
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Invoice ID               1000 non-null   object 
 1   Branch                   1000 non-null   object 
 2   City                     1000 non-null   object 
 3   Customer type            1000 non-null   object 
 4   Gender                   1000 non-null   object 
 5   Product line             1000 non-null   object 
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64  
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object 
 11  Time                     1000 non-null   object 
 12  Payment                  1000 non-null   object 
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

In [7]:

```
df.head()
```

Out[7]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	To
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.97
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.22
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.52
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.04
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.37

In [8]:

```
df.isnull().sum()
```

Out[8]:

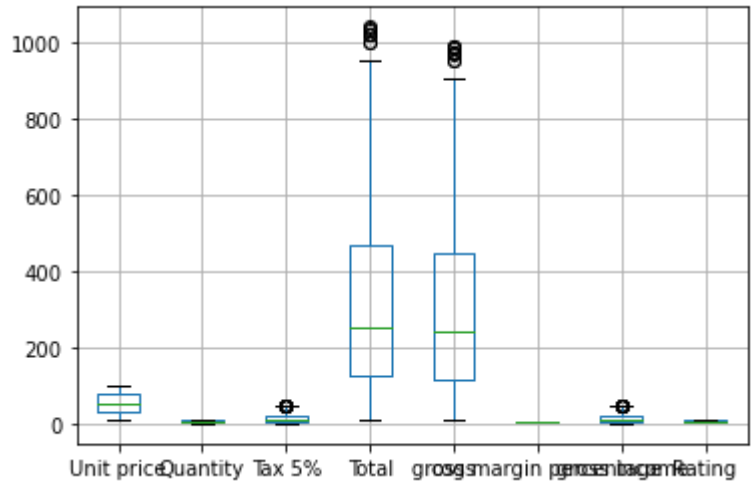
```
Invoice ID                 0
Branch                     0
City                       0
Customer type              0
Gender                     0
Product line               0
Unit price                 0
Quantity                   0
Tax 5%                     0
Total                      0
Date                       0
Time                       0
Payment                    0
cogs                       0
gross margin percentage    0
gross income               0
Rating                     0
dtype: int64
```

In [9]:

```
df.boxplot()
```

Out[9]:

```
<AxesSubplot:>
```



In [10]:

```
df["Customer type"].value_counts()
```

Out[10]:

```
Member    501
Normal    499
Name: Customer type, dtype: int64
```

In [11]:

```
df["Gender"].value_counts()
```

Out[11]:

```
Female    501
Male      499
Name: Gender, dtype: int64
```

In [12]:

```
df["Payment"].value_counts()
```

Out[12]:

```
Ewallet        345
Cash           344
Credit card    311
Name: Payment, dtype: int64
```

In [13]:

```
df["Quantity"].value_counts()
```

Out[13]:

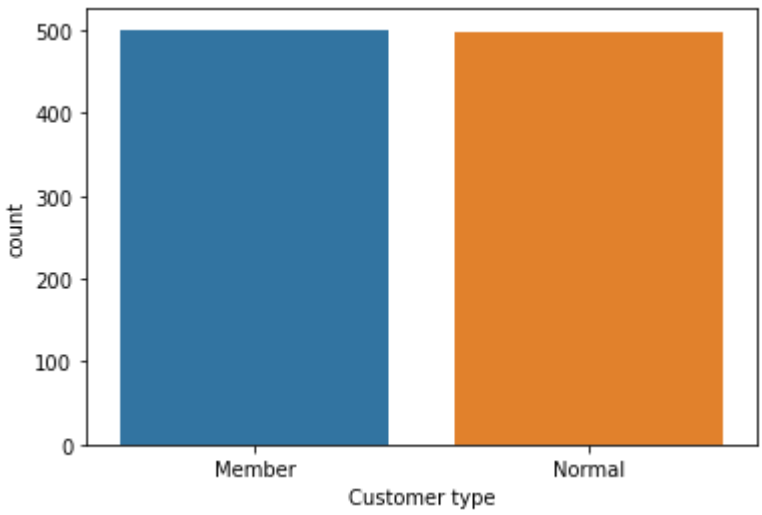
```
10    119
1     112
4     109
7     102
5     102
6      98
9      92
2      91
3      90
8      85
Name: Quantity, dtype: int64
```

In [14]:

```
sns.countplot(x="Customer type",data=df)
```

Out[14]:

```
<AxesSubplot:xlabel='Customer type', ylabel='count'>
```

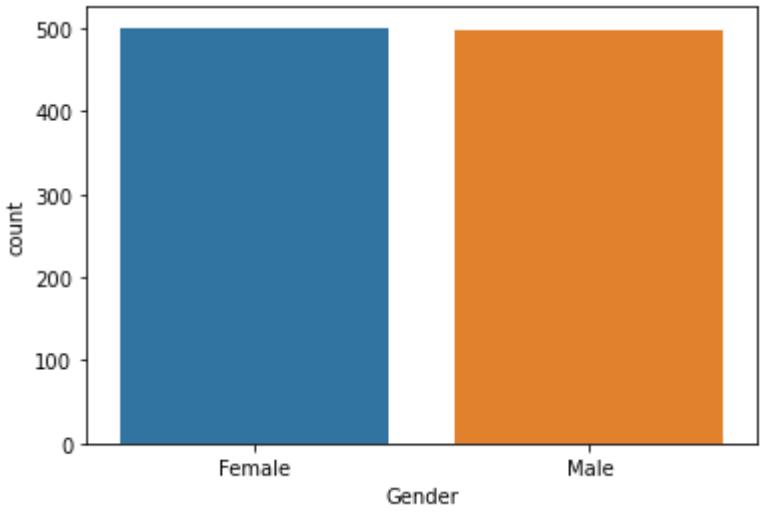


In [15]:

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

Out[15]:

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```

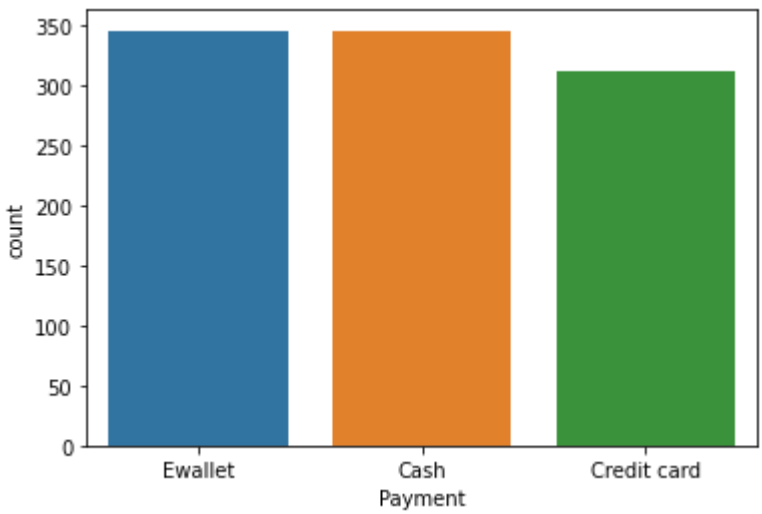


In [16]:

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

Out[16]:

```
<AxesSubplot:xlabel='Payment', ylabel='count'>
```

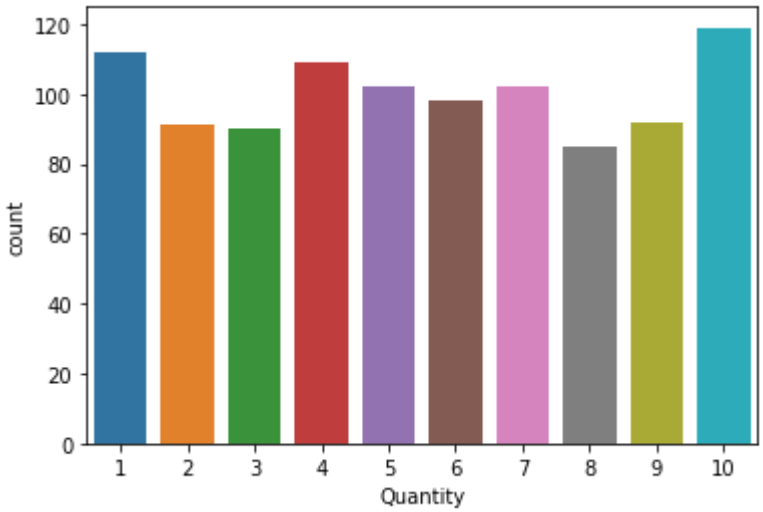


In [17]:

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

Out[17]:

```
<AxesSubplot:xlabel='Quantity', ylabel='count'>
```



In [18]:

```
df.info()
```

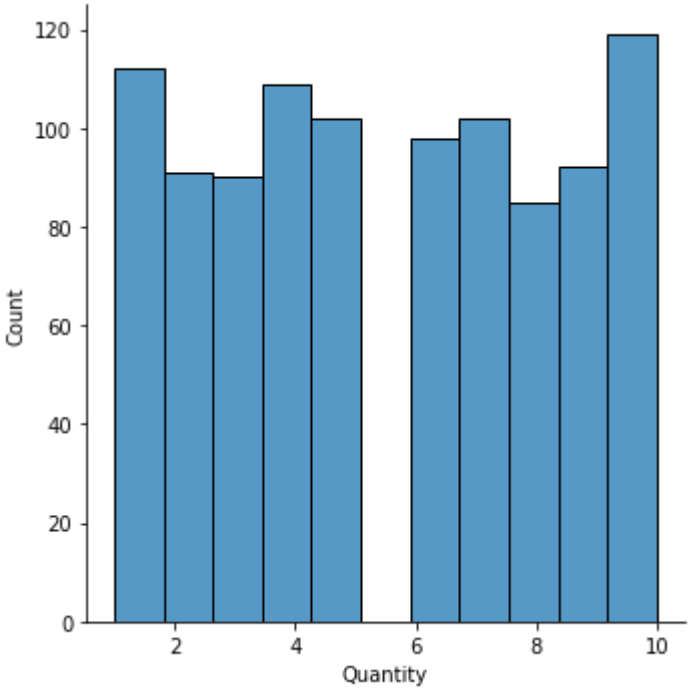
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Invoice ID               1000 non-null   object 
 1   Branch                   1000 non-null   object 
 2   City                     1000 non-null   object 
 3   Customer type            1000 non-null   object 
 4   Gender                   1000 non-null   object 
 5   Product line             1000 non-null   object 
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64  
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object 
 11  Time                     1000 non-null   object 
 12  Payment                  1000 non-null   object 
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

In [19]:

```
sns.displot(df["Quantity"])
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x2cd45aa3340>
```



In [20]:

```
for col in df.columns:
    print('{} : {}'.format(col,df[col].unique()))
```

```
Invoice ID : ['750-67-8428' '226-31-3081' '631-41-3108' '123-19-1176' '373
-73-7910'
 '699-14-3026' '355-53-5943' '315-22-5665' '665-32-9167' '692-92-5582'
 '351-62-0822' '529-56-3974' '365-64-0515' '252-56-2699' '829-34-3910'
 '299-46-1805' '656-95-9349' '765-26-6951' '329-62-1586' '319-50-3348'
 '300-71-4605' '371-85-5789' '273-16-6619' '636-48-8204' '549-59-1358'
 '227-03-5010' '649-29-6775' '189-17-4241' '145-94-9061' '848-62-7243'
 '871-79-8483' '149-71-6266' '640-49-2076' '595-11-5460' '183-56-6882'
 '232-16-2483' '129-29-8530' '272-65-1806' '333-73-7901' '777-82-7220'
 '280-35-5823' '554-53-8700' '354-25-5821' '228-96-1411' '617-15-4209'
 '132-32-9879' '370-41-7321' '727-46-3608' '669-54-1719' '574-22-5561'
 '326-78-5178' '162-48-8011' '616-24-2851' '778-71-5554' '242-55-6721'
 '399-46-5918' '106-35-6779' '635-40-6220' '817-48-8732' '120-06-4233'
 '285-68-5083' '803-83-5989' '347-34-2234' '199-75-8169' '853-23-2453'
 '877-22-3308' '838-78-4295' '109-28-2512' '232-11-3025' '382-03-4532'
 '393-65-2792' '796-12-2025' '510-95-6347' '841-35-6630' '287-21-9091'
 '732-94-0499' '263-10-3913' '381-20-0914' '829-49-1914' '756-01-7507'
 '870-72-4431' '847-38-7188' '480-63-2856' '787-56-0757' '360-39-5055'
 '730-50-9884' '362-58-8315' '633-44-8566' '504-35-8843' '318-68-5053'
```

In [ ]: