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

In [2]:
```python
df = pd.read_csv ( r'C:\Users\USER\Downloads\archive (4)\heart.csv')
df.head(5)
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Attribute Information:

age sex chest pain type (4 values) resting blood pressure serum cholestoral in mg/dl fasting blood sugar > 120 mg/dl resting electrocardiographic results (values 0,1,2) maximum heart rate achieved exercise induced angina oldpeak = ST depression induced by exercise relative to rest the slope of the peak exercise ST segment number of major vessels (0-3) colored by flourosopy thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

In [3]:
```python
df.isnull().sum()
```

Out[3]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [4]:
```python
df.tail()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

In [5]:
```python
df.info
```

Out[5]:
```
<bound method DataFrame.info of       age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0      52    1   0       125   212    0        1      168      0      1.0
1      53    1   0       140   203    1        0      155      1      3.1
2      70    1   0       145   174    0        1      125      1      2.6
3      61    1   0       148   203    0        1      161      0      0.0
4      62    0   0       138   294    1        1      106      0      1.9
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
0         2   2     3       0
1         0   0     3       0
2         0   0     3       0
3         2   1     3       0
4         1   3     2       0
...     ...  ..   ...     ...
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0

[1025 rows x 14 columns]>
```

In [6]:
```python
#Checkng  for duplicate data
df_dup =df.duplicated().any()
print(df_dup)
```

```
True
```

In [7]:
```python
#Dropping duplicated data
df= df.drop_duplicates()
print(df)
```

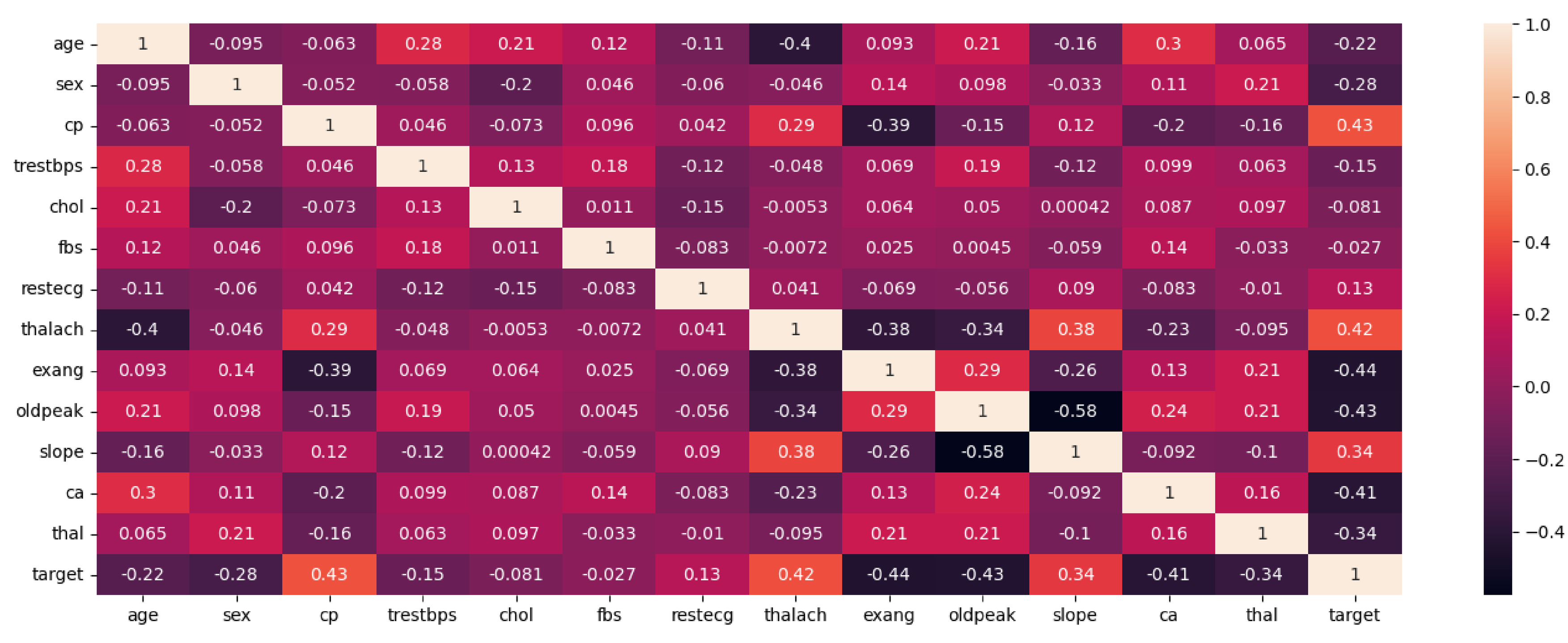
```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     52    1   0       125   212    0        1      168      0      1.0
1     53    1   0       140   203    1        0      155      1      3.1
2     70    1   0       145   174    0        1      125      1      2.6
3     61    1   0       148   203    0        1      161      0      0.0
4     62    0   0       138   294    1        1      106      0      1.9
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
723   68    0   2       120   211    0        0      115      0      1.5
733   44    0   2       108   141    0        1      175      0      0.6
739   52    1   0       128   255    0        1      161      1      0.0
843   59    1   3       160   273    0        0      125      0      0.0
878   54    1   0       120   188    0        1      113      0      1.4

     slope  ca  thal  target
0        2   2     3       0
1        0   0     3       0
2        0   0     3       0
3        2   1     3       0
4        1   3     2       0
..     ...  ..   ...     ...
723      1   0     2       1
733      1   0     2       1
739      2   1     3       0
843      2   0     2       0
878      1   1     3       0

[302 rows x 14 columns]
```

In [8]:
```python
#Getting overall statistics on overall dataset
df.describe()
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	302.00000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	2.314570	0.543046
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.903527	0.470196	1.161452	0.616274	1.006748	0.613026	0.498970
min	29.00000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.00000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.50000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.00000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.00000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

In [9]:
```python
#Drawing Correlation Matrix
plt.figure(figsize=(17,6))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



Number of people with heart diseases and those without

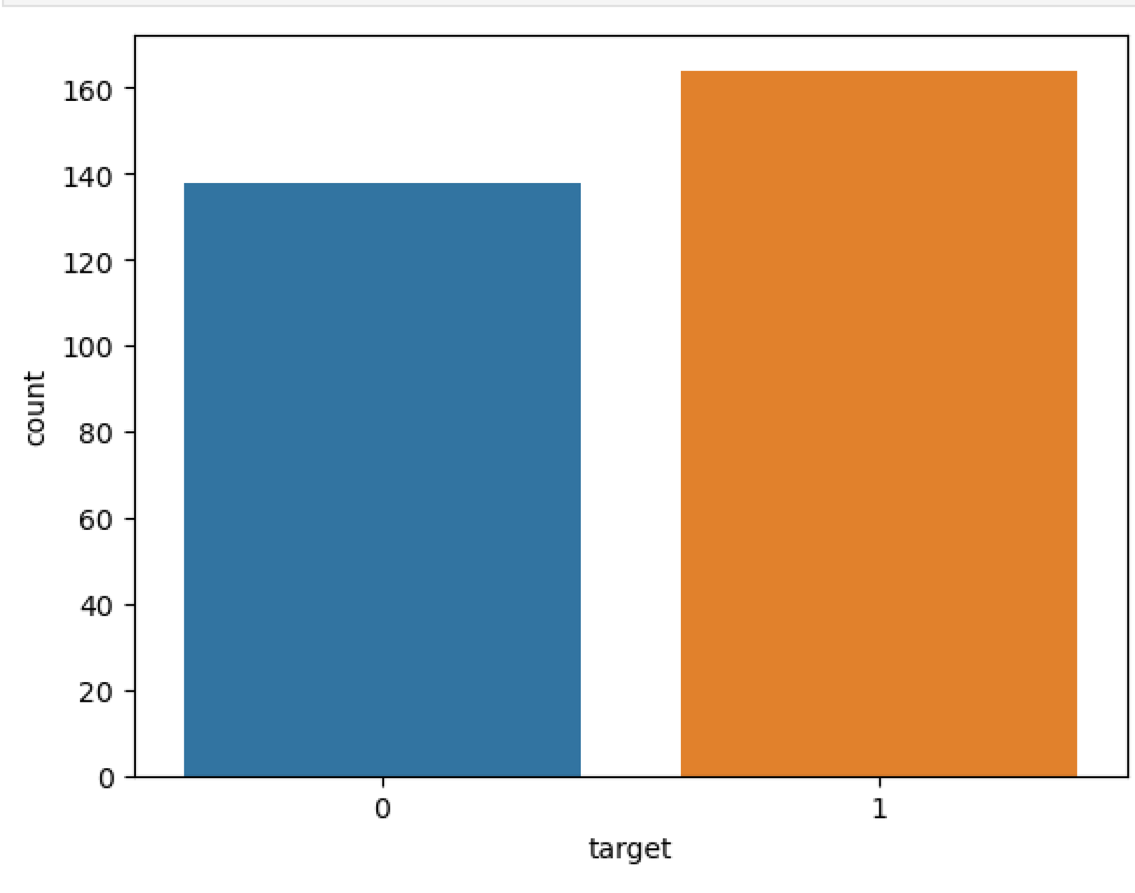
In [10]:
```python
df.columns
```

Out[10]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [11]:
```python
df['target'].value_counts()
```

Out[11]:
```
1    164
0    138
Name: target, dtype: int64
```

In [12]:
```python
sns.countplot(df, x='target')
plt.show()
```



In [ ]:

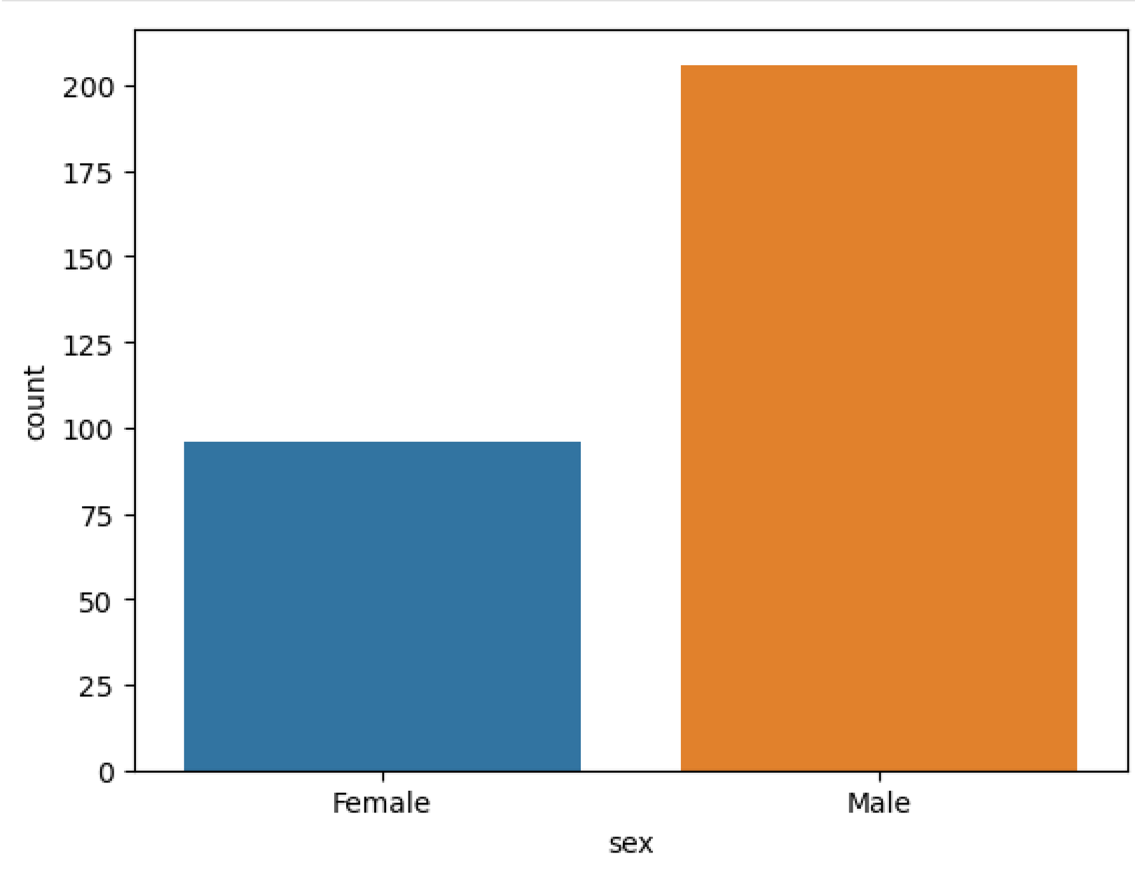
In [ ]:

Find count of Male and Female in the Dataset

In [13]:
```python
df['sex'].value_counts()
```

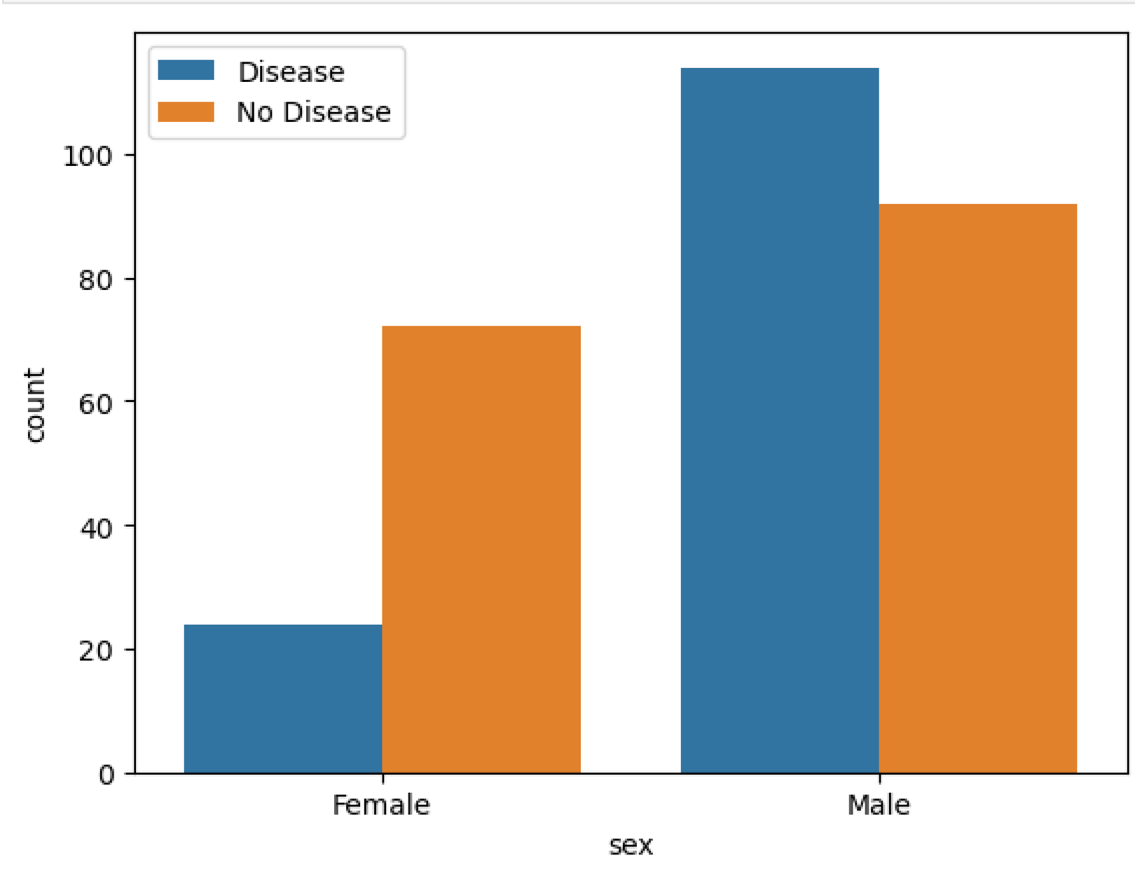
Out[13]:
```
1    206
0     96
Name: sex, dtype: int64
```

In [14]:
```python
sns.countplot(df, x='sex')
plt.xticks([0,1], ['Female' ,'Male'])
plt.show()
```



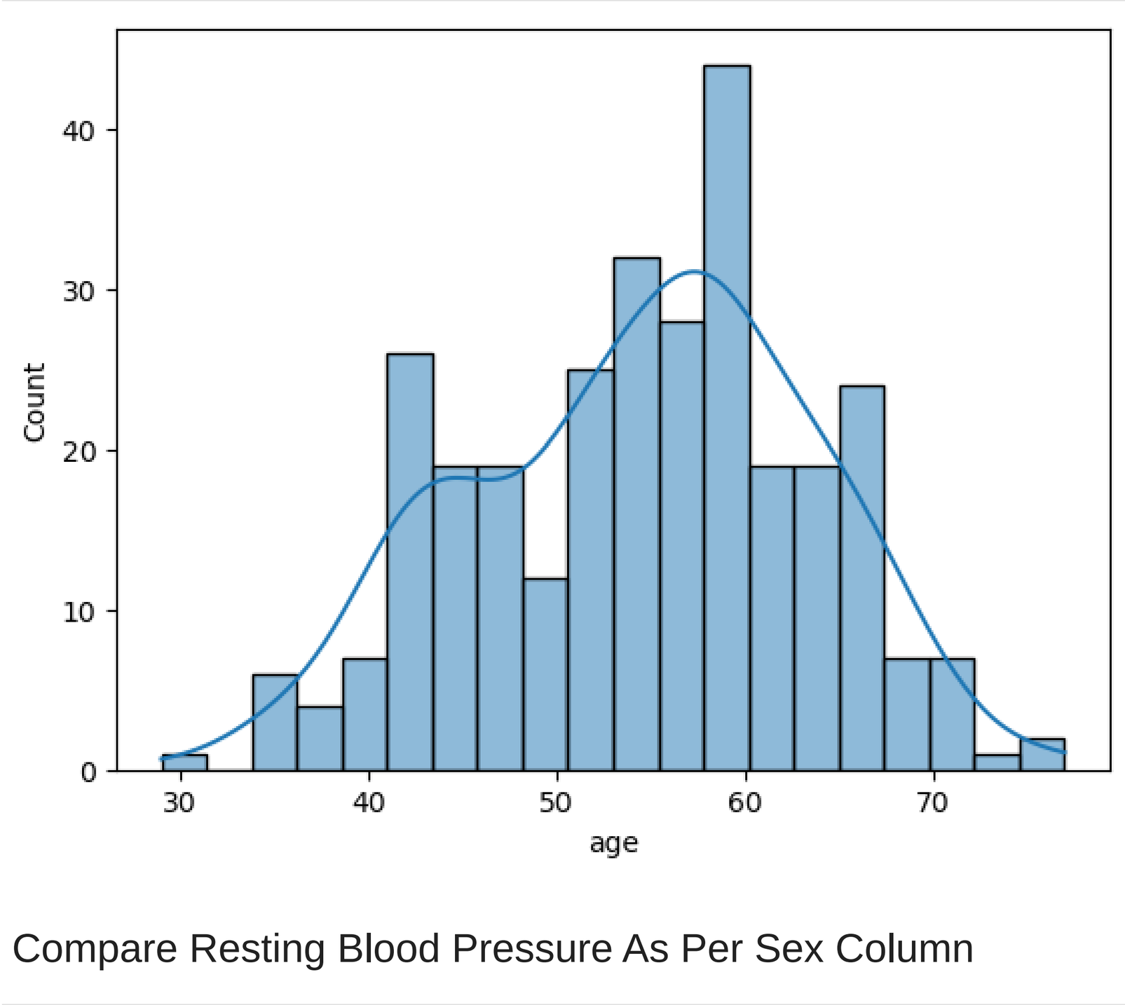
Gender Distribution According to the Target Variable

In [15]:
```python
sns.countplot(df, x='sex', hue='target')
plt.xticks([0,1], ['Female', 'Male'])
plt.legend(labels=['Disease', 'No Disease'])
plt.show()
```



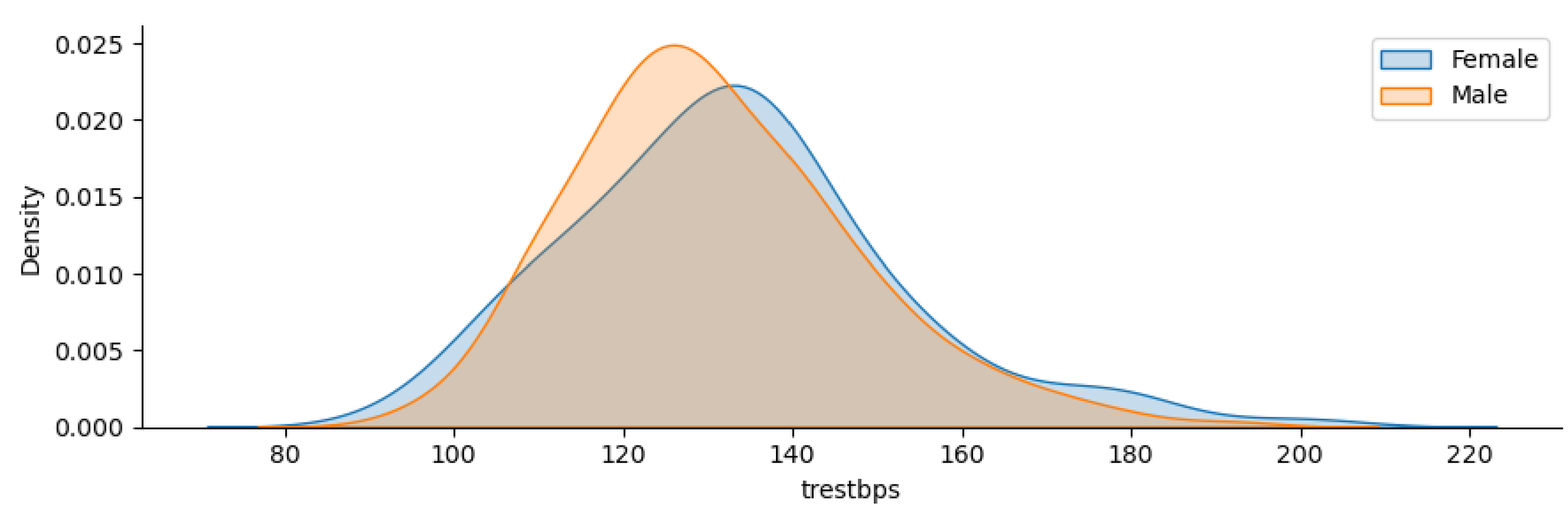
Age Distribution in the Dataset

In [16]:
```python
sns.histplot(df['age'],bins=20, kde=True)
plt.show()
```



Compare Resting Blood Pressure As Per Sex Column

In [17]:
```python
rbp= sns.FacetGrid(df, hue='sex', aspect= 3)
rbp.map(sns.kdeplot,'trestbps',fill=True)
plt.legend(labels=['Female', 'Male'])
plt.show()
```



Plotting for Continous Variables

In [18]:
```python
df.columns
```

Out[18]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [19]:
```python
cate_val=[]
cont_val=[]

for column in df.columns:
    if df[column].nunique () <=10:
        cate_val.append(column)
    else:
        cont_val.append(column)
```

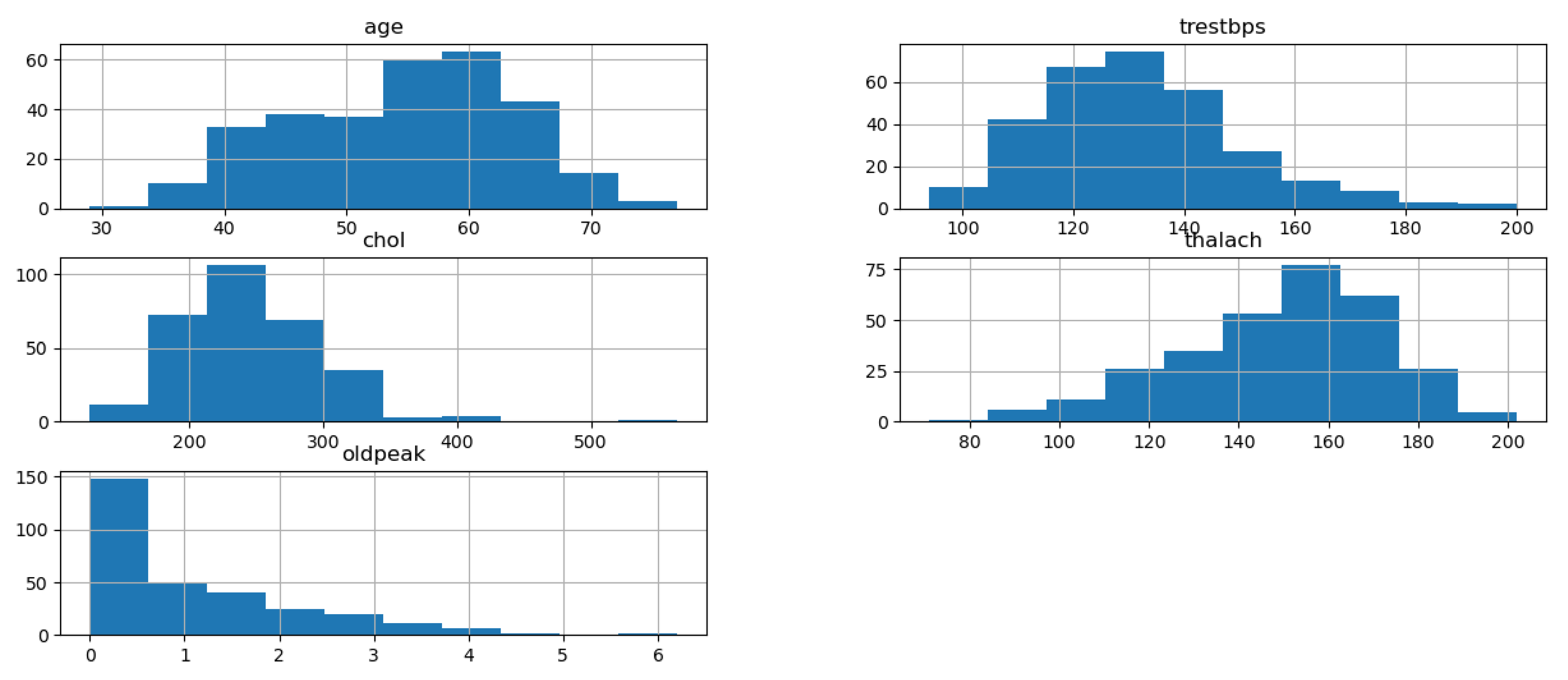
In [20]:
```python
cate_val
```

Out[20]:
```
['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

In [21]:
```python
cont_val
```

Out[21]:
```
['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

In [22]:
```python
df.hist(cont_val, figsize=[15,6])
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: