Importing libraries and dataset and explore rows

In [19]:

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

In [20]:

```
heart=pd.read_csv("D:\\Career\\Unified_Mentor_Intenship\\Heart Disease data\\He
heart.head(5)
```

Out[20]:

	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

Checking The Shape of the Dataset

In [21]:

```
print('The number of rows in the dataset ',heart.shape[0])
print('The number of columns in the dataset ',heart.shape[1])
```

```
The number of rows in the dataset  1025
The number of columns in the dataset  14
```

The detailing of the dataset

In [22]:

```
heart.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

This indicates there is no null values present in the dataset

Checking duplicates in the data

In [23]:

```
heart.duplicated().sum()
```

Out[23]:

```
723
```

As this data has no uniqe id identifying different patient, so there will be possiblity for duplicates in this data with different patients. So Keeping all duplicates for further analysis

Get overall Statistics of the dataset

In [24]:
```
heart.describe(include='all')
```

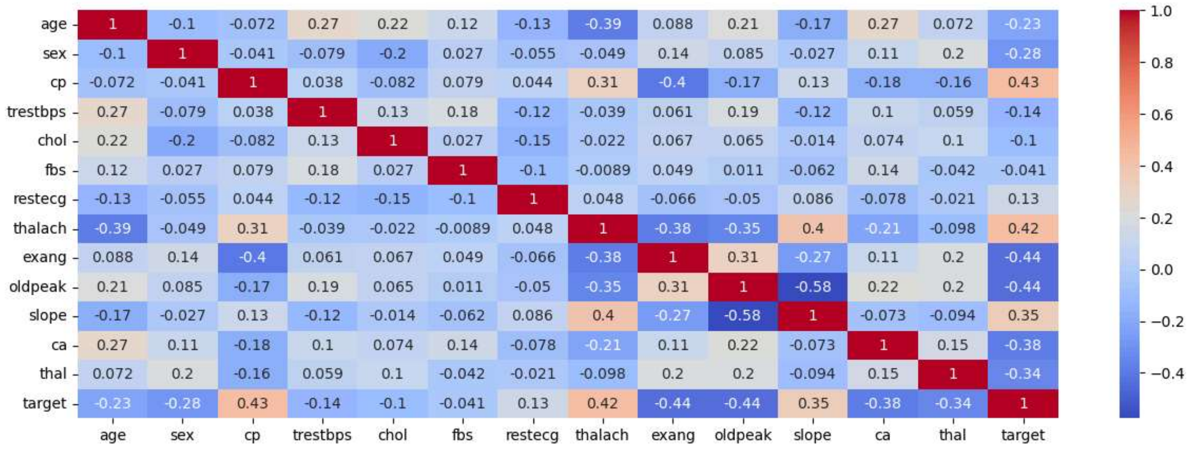
Out[24]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000

Draw Correlation Matrix

In [25]:
```
cor=heart.corr()
```

In [26]:
```
plt.figure(figsize=(15,5))
sns.heatmap(cor,annot=True,cmap='coolwarm')
plt.show()
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.1	-0.072	0.27	0.22	0.12	-0.13	-0.39	0.088	0.21	-0.17	0.27	0.072	-0.23
sex	-0.1	1	-0.041	-0.079	-0.2	0.027	-0.055	-0.049	0.14	0.085	-0.027	0.11	0.2	-0.28
cp	-0.072	-0.041	1	0.038	-0.082	0.079	0.044	0.31	-0.4	-0.17	0.13	-0.18	-0.16	0.43
trestbps	0.27	-0.079	0.038	1	0.13	0.18	-0.12	-0.039	0.061	0.19	-0.12	0.1	0.059	-0.14
chol	0.22	-0.2	-0.082	0.13	1	0.027	-0.15	-0.022	0.067	0.065	-0.014	0.074	0.1	-0.1
fbs	0.12	0.027	0.079	0.18	0.027	1	-0.1	-0.0089	0.049	0.011	-0.062	0.14	-0.042	-0.041
restecg	-0.13	-0.055	0.044	-0.12	-0.15	-0.1	1	0.048	-0.066	-0.05	0.086	-0.078	-0.021	0.13
thalach	-0.39	-0.049	0.31	-0.039	-0.022	-0.0089	0.048	1	-0.38	-0.35	0.4	-0.21	-0.098	0.42
exang	0.088	0.14	-0.4	0.061	0.067	0.049	-0.066	-0.38	1	0.31	-0.27	0.11	0.2	-0.44
oldpeak	0.21	0.085	-0.17	0.19	0.065	0.011	-0.05	-0.35	0.31	1	-0.58	0.22	0.2	-0.44
slope	-0.17	-0.027	0.13	-0.12	-0.014	-0.062	0.086	0.4	-0.27	-0.58	1	-0.073	-0.094	0.35
ca	0.27	0.11	-0.18	0.1	0.074	0.14	-0.078	-0.21	0.11	0.22	-0.073	1	0.15	-0.38
thal	0.072	0.2	-0.16	0.059	0.1	-0.042	-0.021	-0.098	0.2	0.2	-0.094	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.1	-0.041	0.13	0.42	-0.44	-0.44	0.35	-0.38	-0.34	1

Heart Disease count (Presence or no Presence)?

In [27]:
```
heart['target'].value_counts()
```

Out[27]:
```
target
1    526
0    499
Name: count, dtype: int64
```

This indicates heart disease presence is higher than individuals with no heart disease

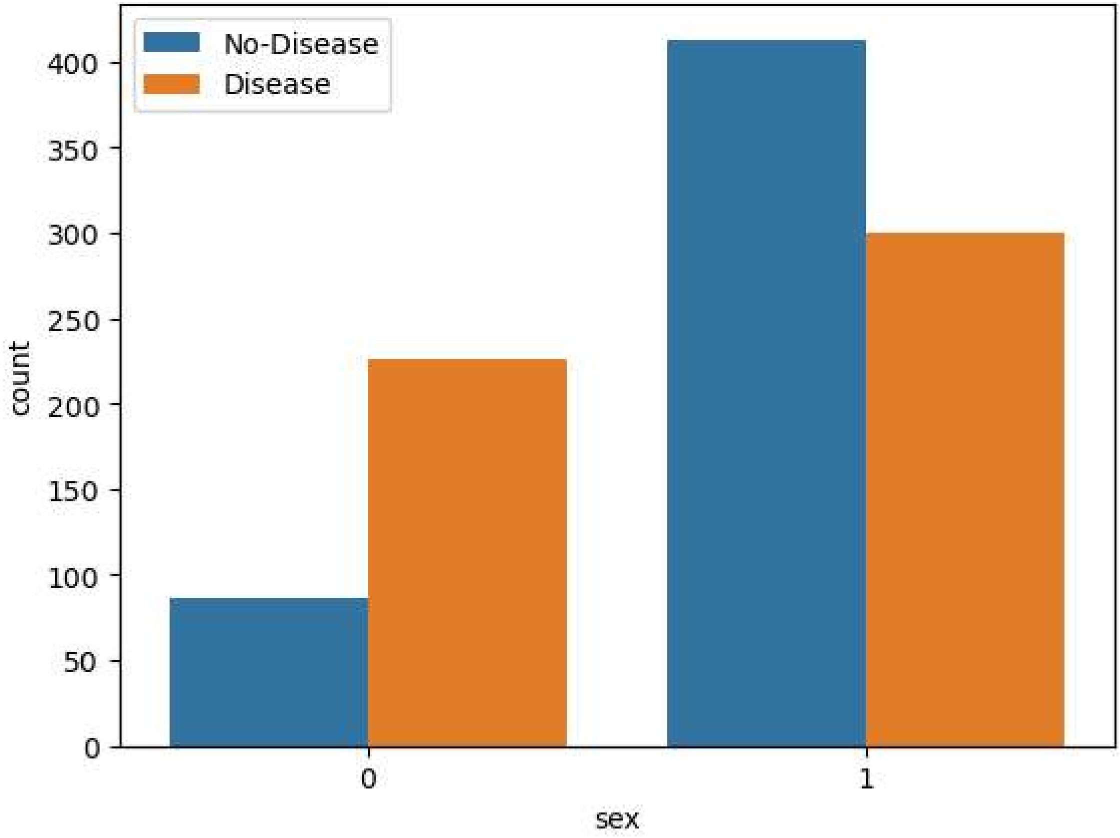
Which sex has the most heart disease (Male or Female)?

In [28]:

```
sns.countplot(x='sex',data=heart,hue='target')
plt.legend(labels=['No-Disease','Disease'])
```

Out[28]:

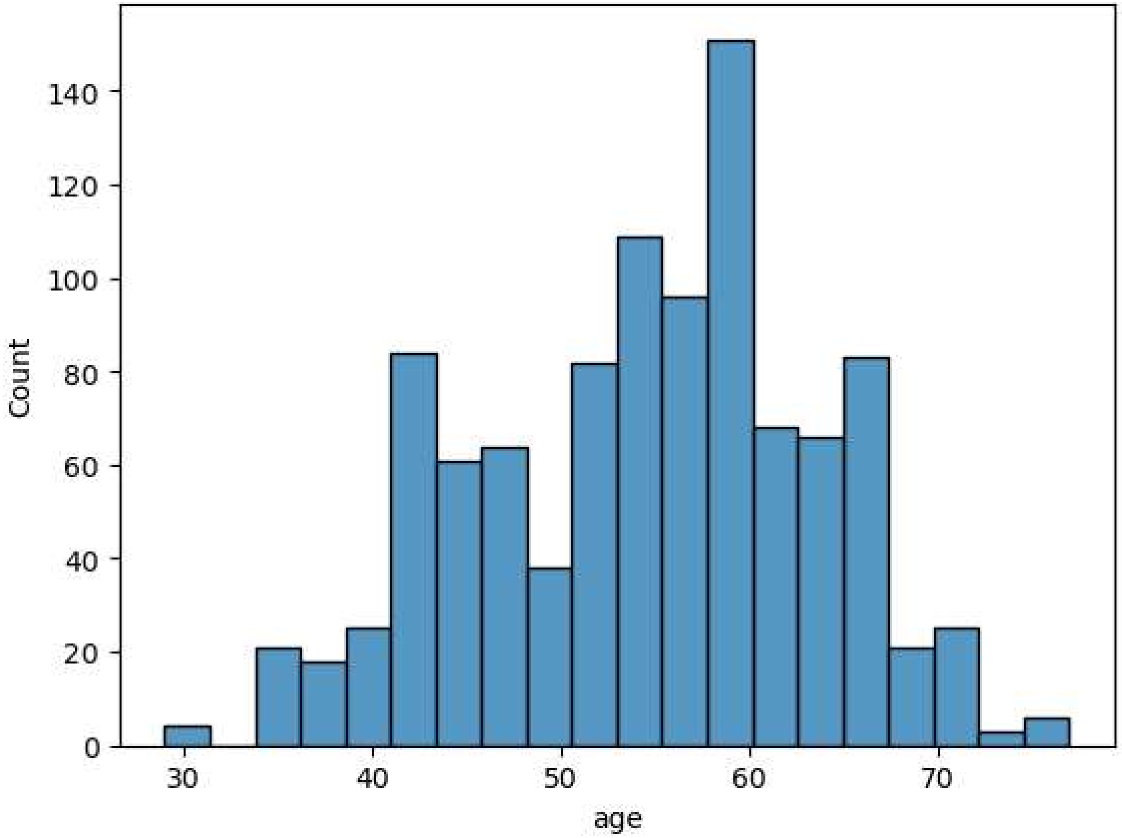
```
<matplotlib.legend.Legend at 0x1db572ff910>
```



It is clearly indicates that male (sex value 1) have higer presence of heart disease comapre to female (sex value 0)

Check Age Distribution in the Dataset

In [29]:
```
sns.histplot(heart['age'],bins=20)
plt.show()
```



Check Chest Pain Type

Chest pain type (4 values)

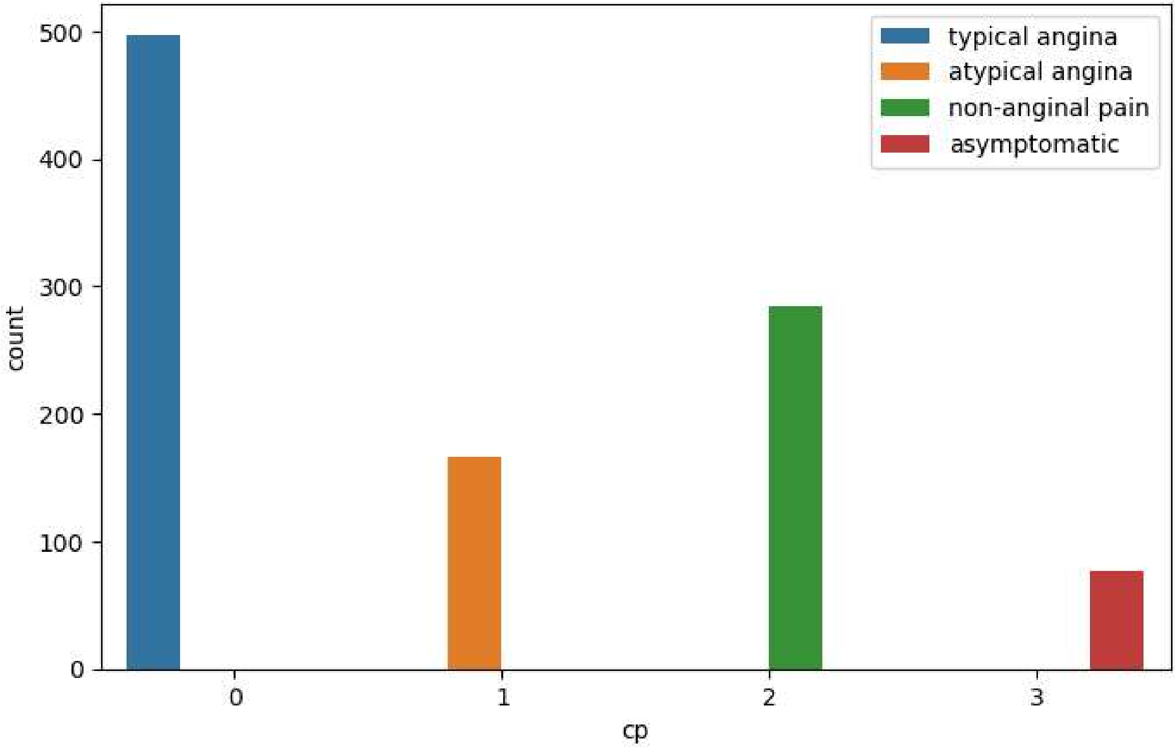
0:typical angina

1:atypical angina

2:non-anginal pain

3:asymptomatic

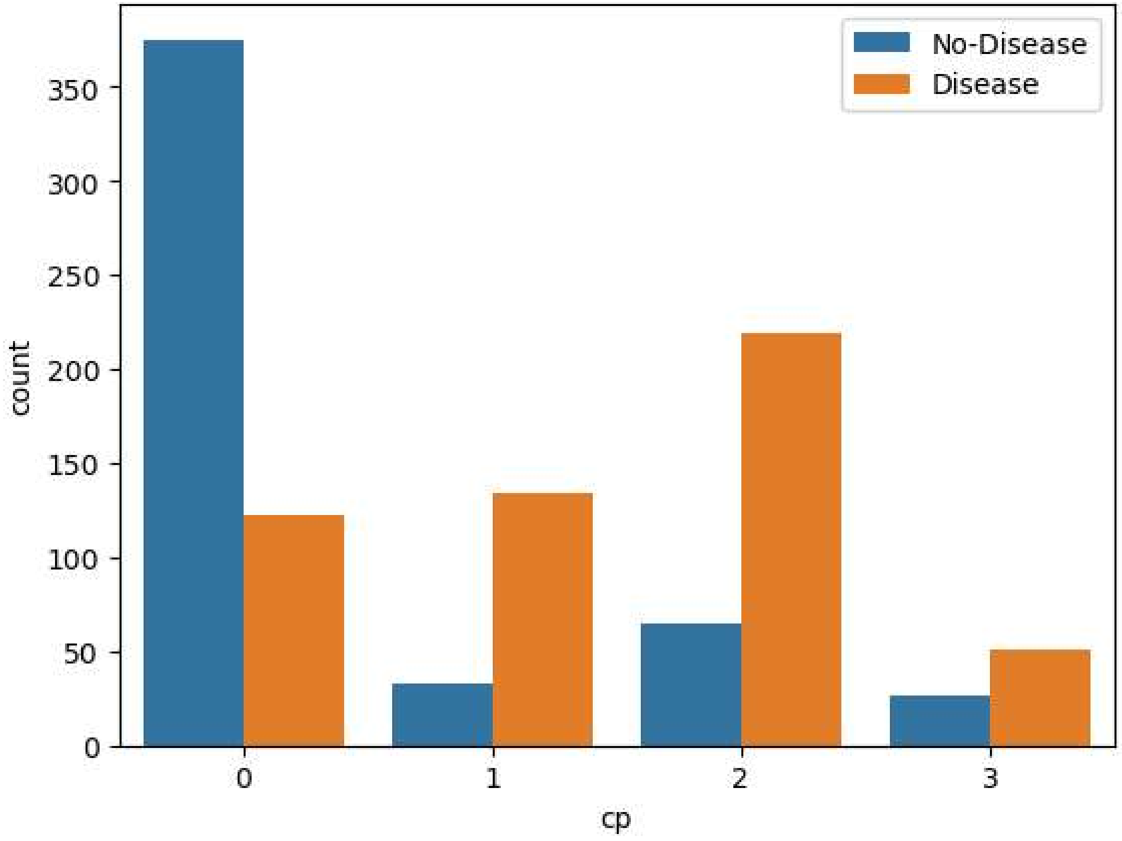
```
In [30]: plt.figure(figsize=(8,5))
         sns.countplot(x='cp',data=heart,hue='cp')
         plt.legend(labels=['typical angina','atypical angina','non-anginal pain','asymp
         plt.show()
```



Chest Pain Distribution as per Target Variable

In [31]:

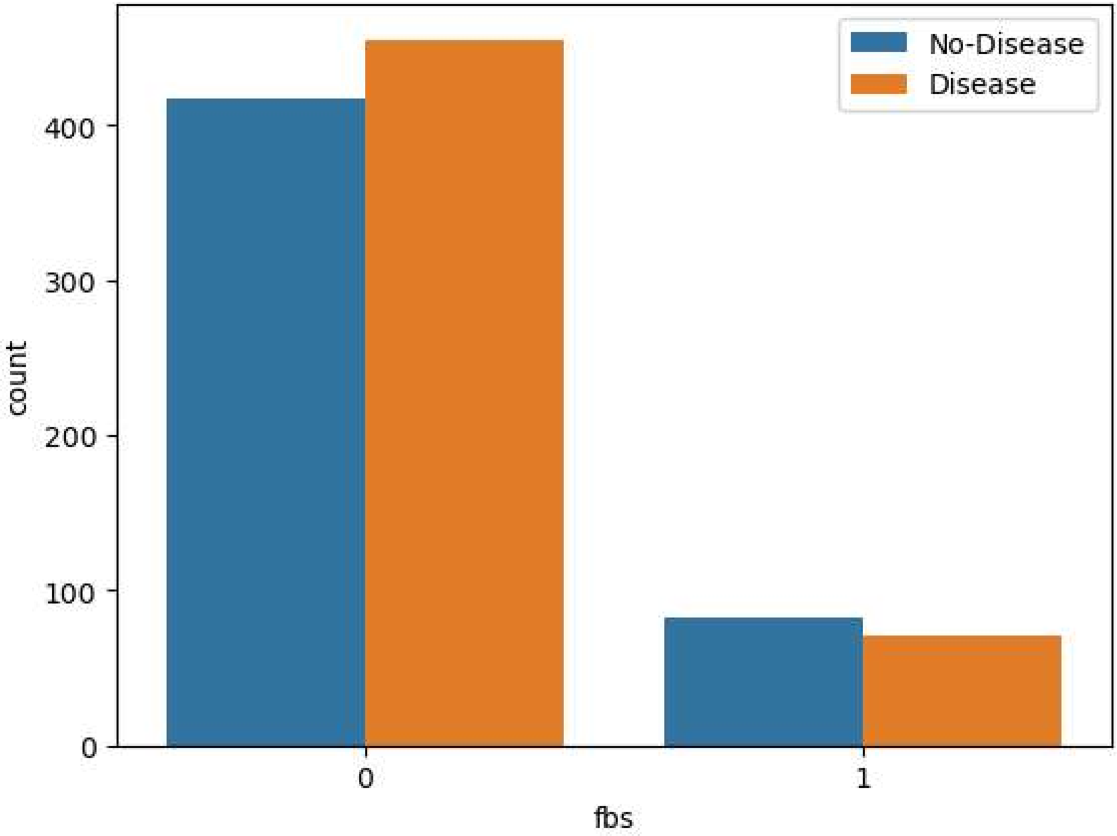
```
sns.countplot(x='cp',data=heart,hue='target')
plt.legend(labels=['No-Disease','Disease'])
plt.show()
```



Fasting Blood Sugar Distribution According to the Target Variable

In [32]:

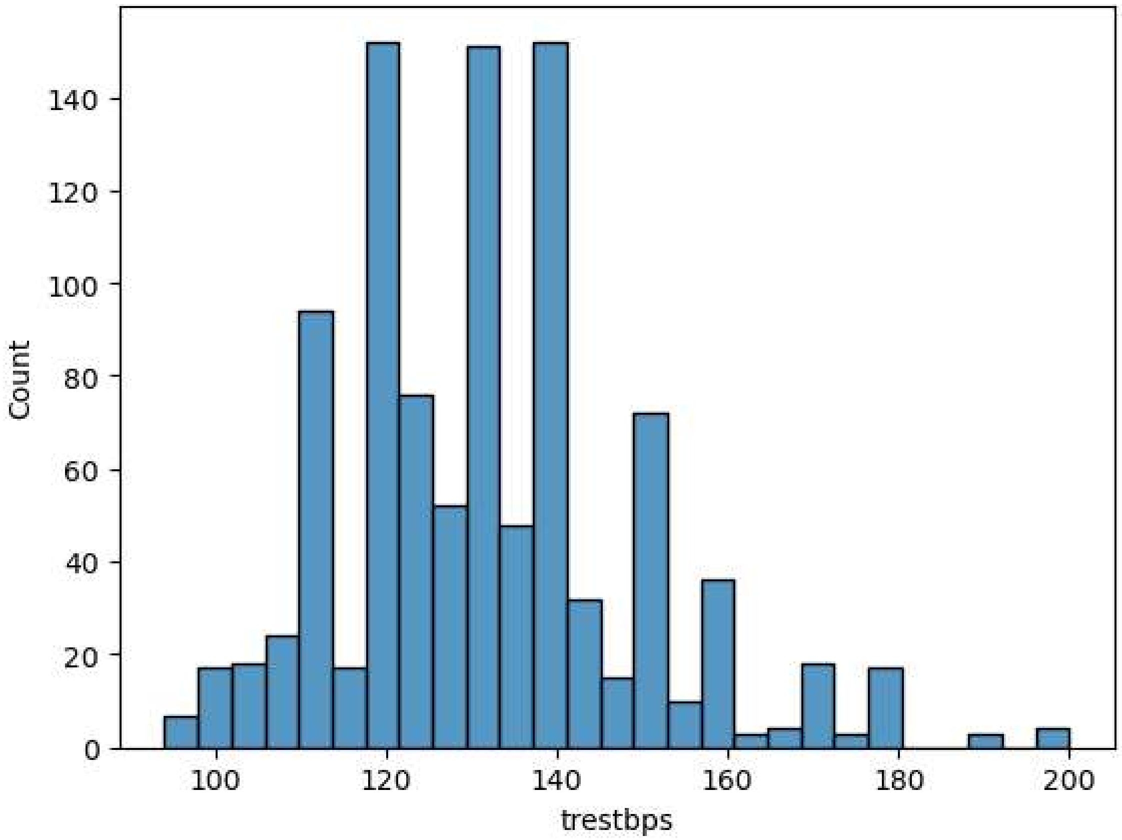
```
sns.countplot(x='fbs',data=heart,hue='target')
plt.legend(labels=['No-Disease','Disease'])
plt.show()
```



Check Resting Blood Pressure Distribution

In [33]:
```
sns.histplot(heart['trestbps'])
```

Out[33]:
```
<Axes: xlabel='trestbps', ylabel='Count'>
```

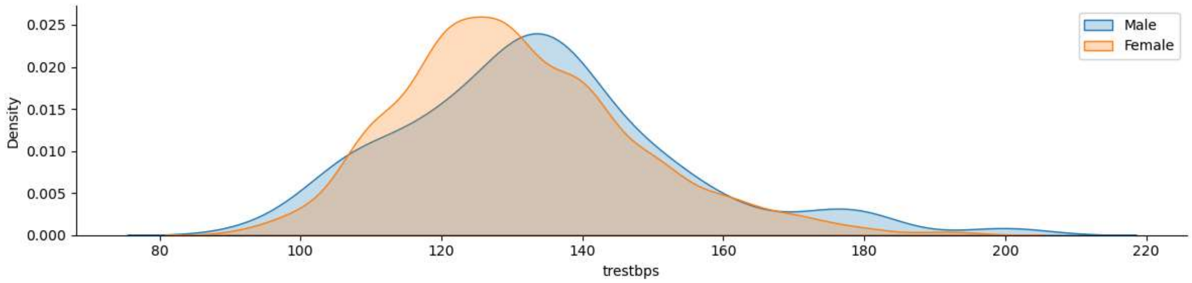


Comparing Resting Blood Pressure with Sex

In [37]:
```
f=sns.FacetGrid(heart,hue='sex',aspect=4)
f.map(sns.kdeplot,'trestbps',fill=True)
plt.legend(labels=['Male','Female'])
```

```
d:\files\Anaconda\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The
figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

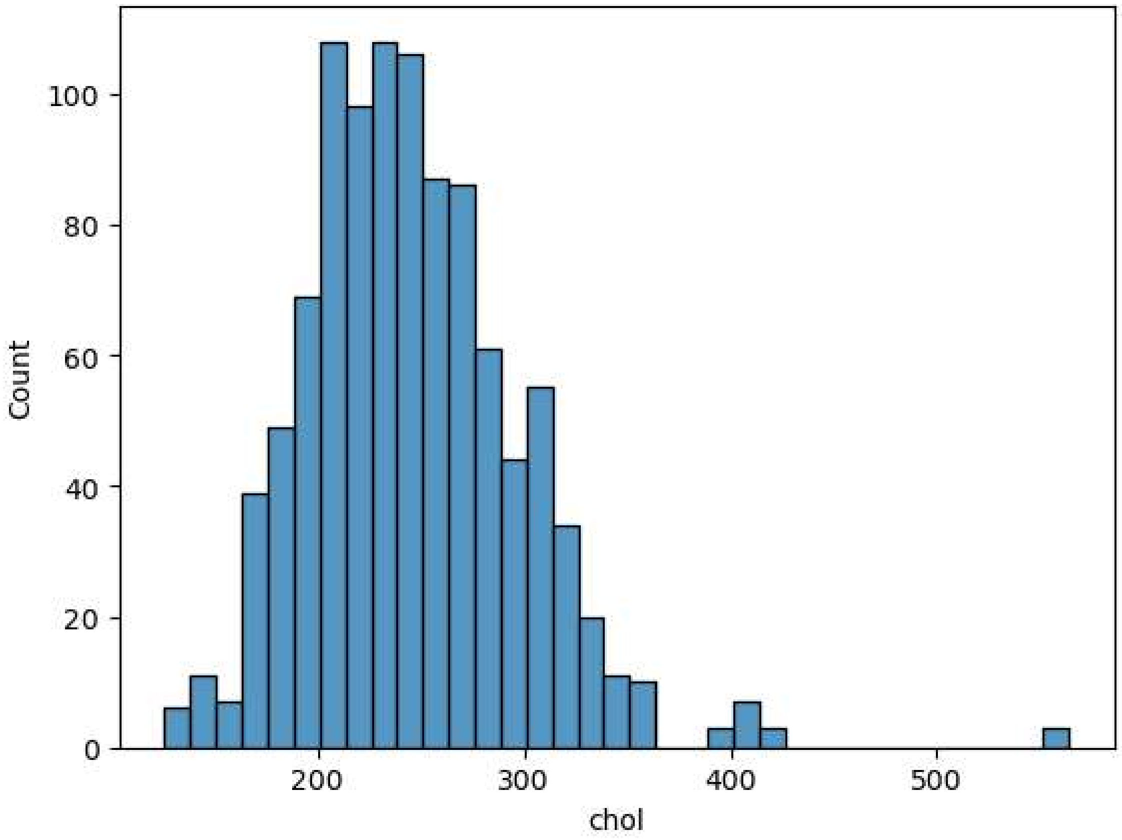
Out[37]:
```
<matplotlib.legend.Legend at 0x1db572c6850>
```



Distribution of Serum Cholesterol

```
In [38]: sns.histplot(heart['chol'])
```

```
Out[38]: <Axes: xlabel='chol', ylabel='Count'>
```



```
In [ ]:
```