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

In [2]:
```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data'
columns = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang','oldpeak', 'slope', 'ca
data = pd.read_csv(url, names=columns)
```

In [3]:
```
data['target'] = data['target'] .apply(lambda x: 1 if x > 0 else 0)#convert the the values in target greater tha
data
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	1
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45.0	1.0	1.0	110.0	264.0	0.0	0.0	132.0	0.0	1.2	2.0	0.0	7.0	1
299	68.0	1.0	4.0	144.0	193.0	1.0	0.0	141.0	0.0	3.4	2.0	2.0	7.0	1
300	57.0	1.0	4.0	130.0	131.0	0.0	0.0	115.0	1.0	1.2	2.0	1.0	7.0	1
301	57.0	0.0	2.0	130.0	236.0	0.0	2.0	174.0	0.0	0.0	2.0	1.0	3.0	1
302	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	?	3.0	0

303 rows × 14 columns

In [4]:
```
missing=data[data.isin(['?']).any(axis=1)]
missing
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
87	53.0	0.0	3.0	128.0	216.0	0.0	2.0	115.0	0.0	0.0	1.0	0.0	?	0
166	52.0	1.0	3.0	138.0	223.0	0.0	0.0	169.0	0.0	0.0	1.0	?	3.0	0
192	43.0	1.0	4.0	132.0	247.0	1.0	2.0	143.0	1.0	0.1	2.0	?	7.0	1
266	52.0	1.0	4.0	128.0	204.0	1.0	0.0	156.0	1.0	1.0	2.0	0.0	?	1
287	58.0	1.0	2.0	125.0	220.0	0.0	0.0	144.0	0.0	0.4	2.0	?	7.0	0
302	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	?	3.0	0

In [5]:
```
data['ca']=data['ca'].replace('?',float("nan"))#replace the value ? to nan and filling the values the mean of t
data['ca']=pd.to_numeric(data['ca'])
data['ca']=data['ca'].fillna(data['ca'].mean())
data
```

Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.000000	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.000000	3.0	1
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.000000	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.000000	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.000000	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45.0	1.0	1.0	110.0	264.0	0.0	0.0	132.0	0.0	1.2	2.0	0.000000	7.0	1
299	68.0	1.0	4.0	144.0	193.0	1.0	0.0	141.0	0.0	3.4	2.0	2.000000	7.0	1
300	57.0	1.0	4.0	130.0	131.0	0.0	0.0	115.0	1.0	1.2	2.0	1.000000	7.0	1
301	57.0	0.0	2.0	130.0	236.0	0.0	2.0	174.0	0.0	0.0	2.0	1.000000	3.0	1
302	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	0.672241	3.0	0

303 rows × 14 columns

In [6]:
```
missing=data[data.isin(['?']).any(axis=1)]
missing
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
87	53.0	0.0	3.0	128.0	216.0	0.0	2.0	115.0	0.0	0.0	1.0	0.0	?	0
266	52.0	1.0	4.0	128.0	204.0	1.0	0.0	156.0	1.0	1.0	2.0	0.0	?	1

In [7]:
```
data.replace('?',np.nan,inplace=True)#droping the rows with ? value in it
data.dropna(inplace=True)
data
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.000000	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.000000	3.0	1
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.000000	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.000000	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.000000	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45.0	1.0	1.0	110.0	264.0	0.0	0.0	132.0	0.0	1.2	2.0	0.000000	7.0	1
299	68.0	1.0	4.0	144.0	193.0	1.0	0.0	141.0	0.0	3.4	2.0	2.000000	7.0	1
300	57.0	1.0	4.0	130.0	131.0	0.0	0.0	115.0	1.0	1.2	2.0	1.000000	7.0	1
301	57.0	0.0	2.0	130.0	236.0	0.0	2.0	174.0	0.0	0.0	2.0	1.000000	3.0	1
302	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	0.672241	3.0	0

301 rows × 14 columns

In [8]:
```
missing=data[data.isin(['?']).any(axis=1)]
missing
```

Out[8]:

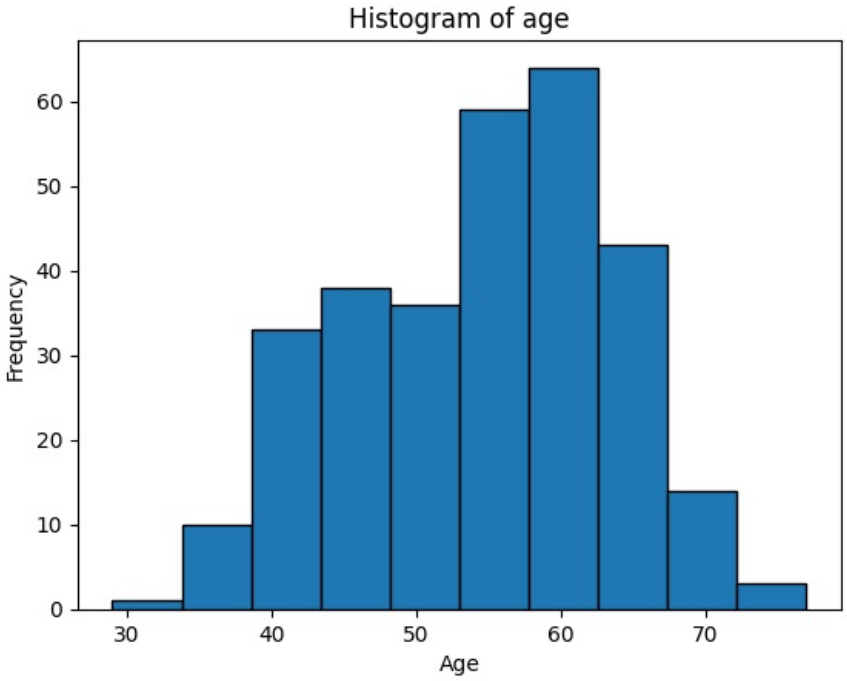
age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target

In [9]:
```
descriptive_stats = data.describe()#Generating descriptive statistics for the dataset to  understand the distri
descriptive_stats
```

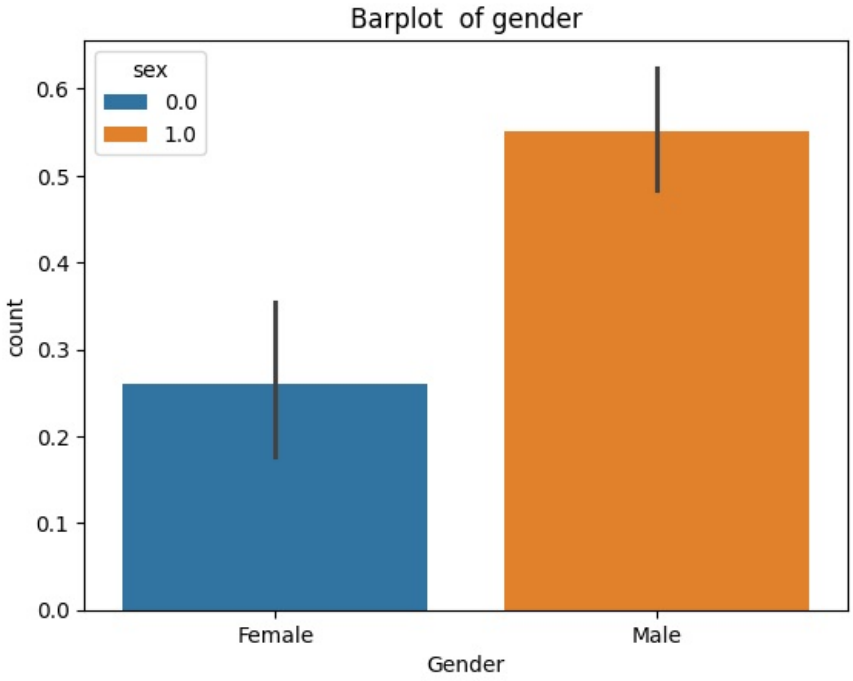
Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	
count	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	3
mean	54.451827	0.681063	3.156146	131.714286	246.936877	0.146179	0.990033	149.700997	0.325581	1.043189	
std	9.067258	0.466841	0.962048	17.655729	51.859869	0.353874	0.994937	22.860817	0.469372	1.163384	
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	
25%	48.000000	0.000000	3.000000	120.000000	211.000000	0.000000	0.000000	134.000000	0.000000	0.000000	
50%	56.000000	1.000000	3.000000	130.000000	242.000000	0.000000	1.000000	153.000000	0.000000	0.800000	
75%	61.000000	1.000000	4.000000	140.000000	275.000000	0.000000	2.000000	166.000000	1.000000	1.600000	
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	

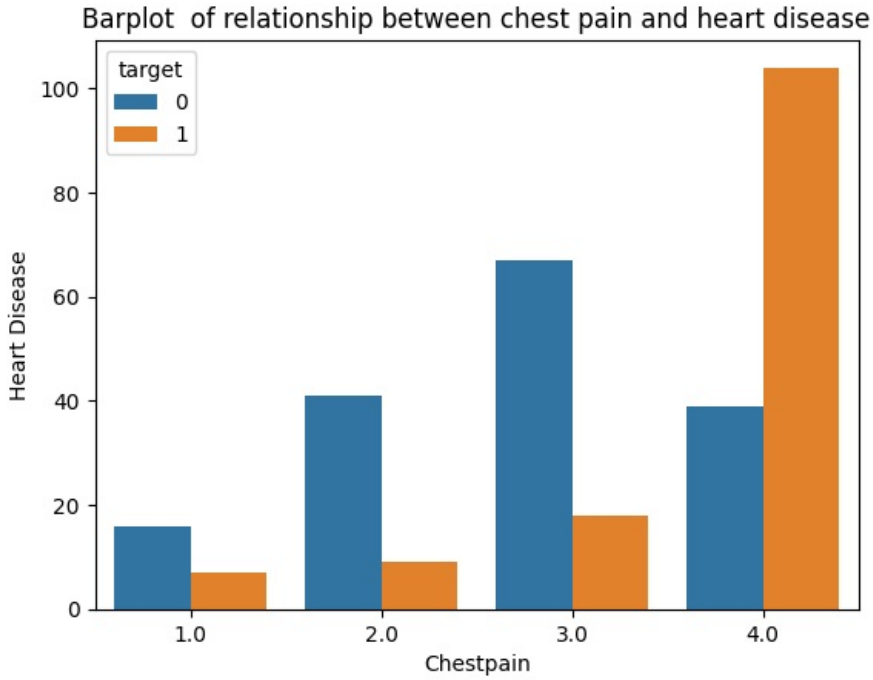
In [10]:
```
plt.hist(data["age"],edgecolor="black")#histogram to visualize the distribution of ages in the dataset.
plt.title("Histogram of age")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()
```



In [11]:
```
sns.barplot(data=data, x="sex",y="target" ,hue='sex')#bar plot to visualize the distribution of gender in the da
plt.xticks(ticks=[0,1],labels=["Female","Male"])
plt.title("Barplot  of gender")
plt.xlabel("Gender")
plt.ylabel("count")
plt.show()
```

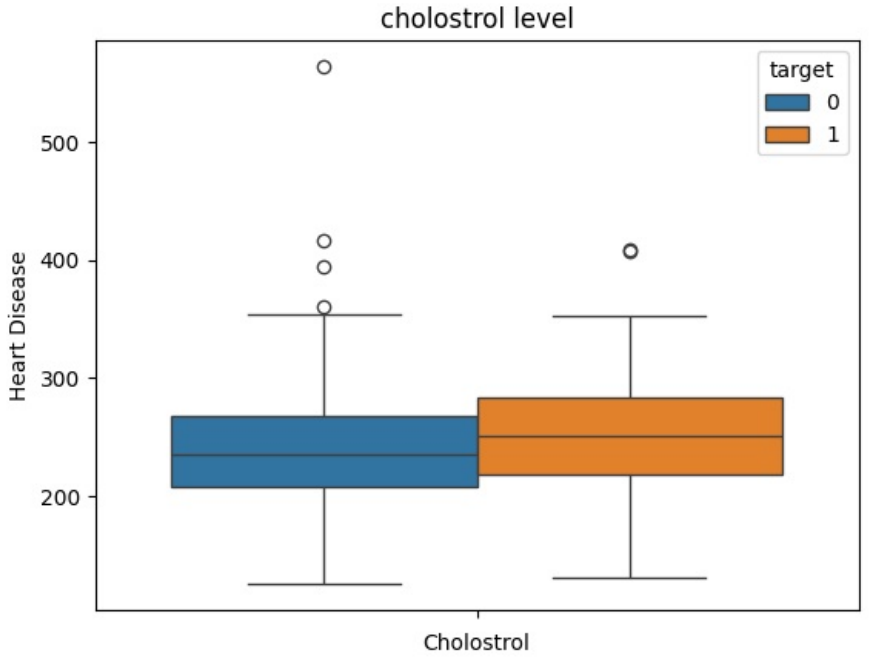


In [12]:
```
sns.countplot(data=data,x="cp",hue="target")#count plot to visualize the relationship between chest pain and th
plt.title("Barplot  of relationship between chest pain and heart disease")
plt.xlabel("Chestpain")
plt.ylabel("Heart Disease")
plt.show()
```

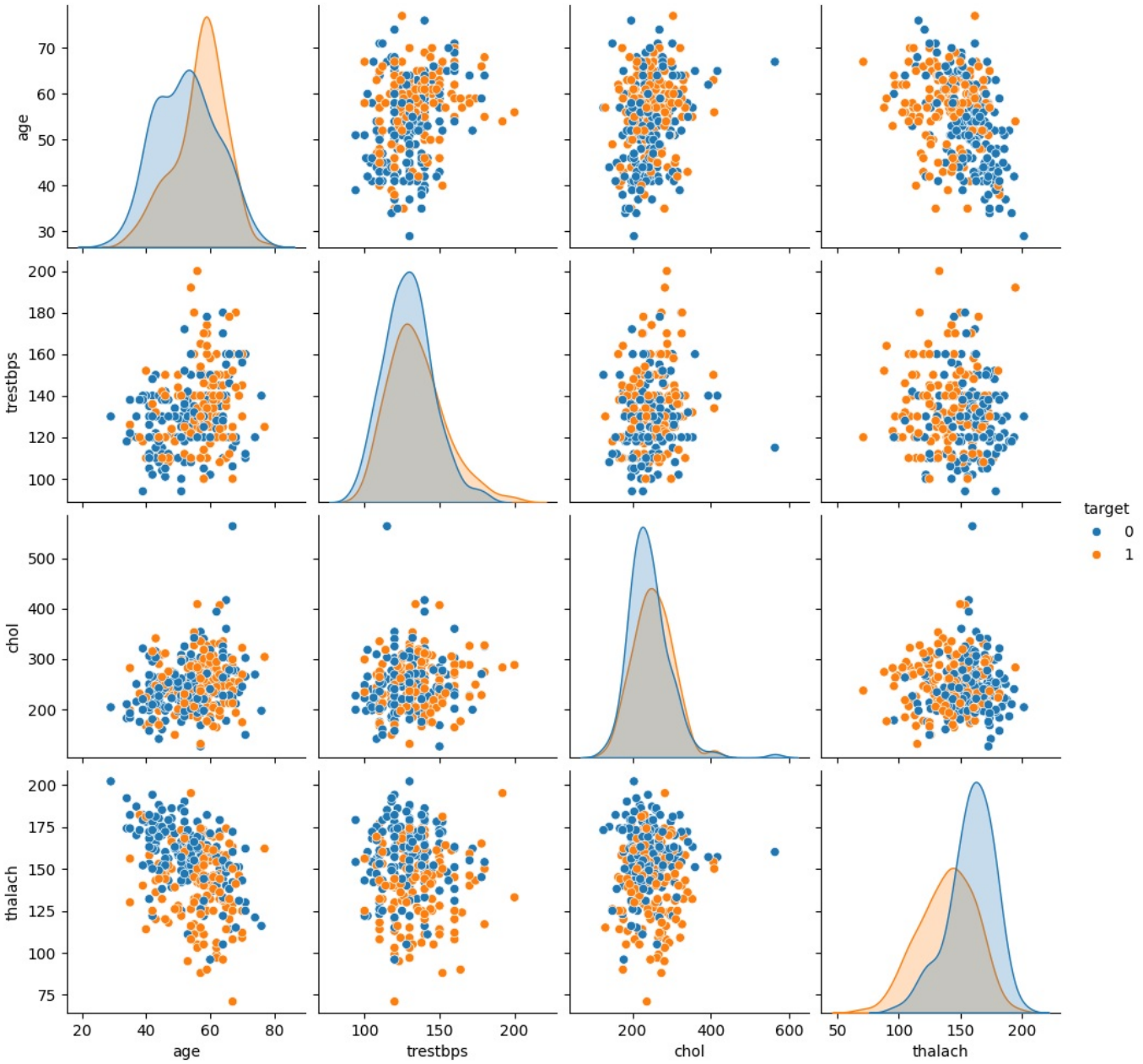


In [18]:
```
sns.boxplot(data=data,hue='target',y='chol')#Box plot to visualize the distribution of cholesterol levels  for 
plt.title('cholostrol level')
plt.xlabel("Cholostrol")
plt.ylabel("Heart Disease")

plt.show()
```

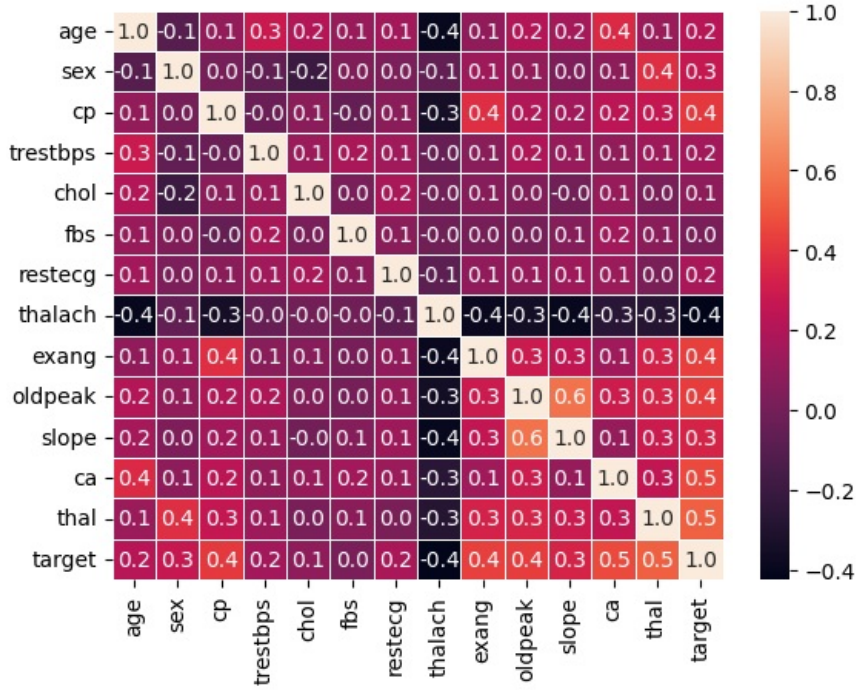


In [14]:
```
sns.pairplot(data=data[['age', 'trestbps', 'chol', 'thalach' ,'target' ]],hue='target')#Pair plot to visualize 
plt.show()
```

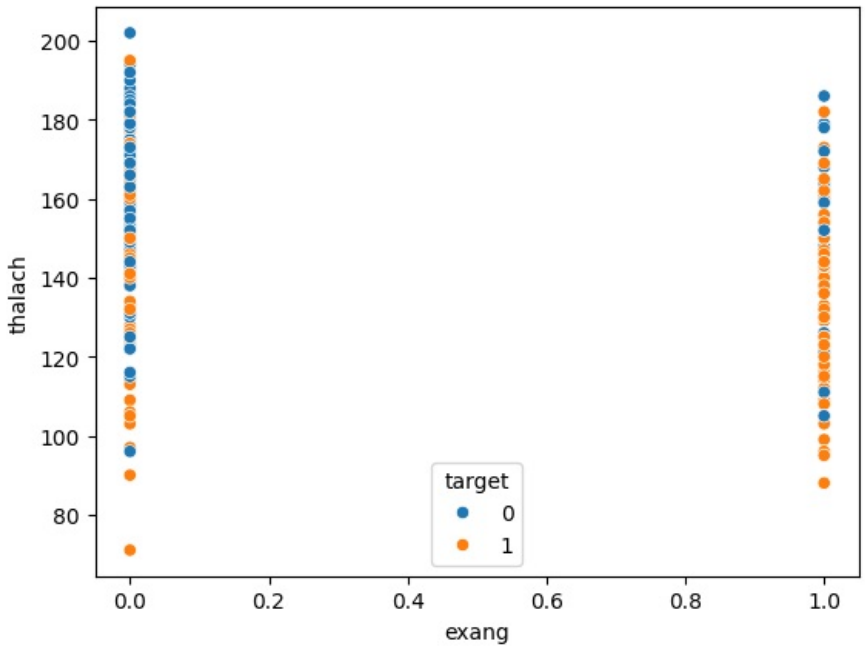


In [15]:
```
sns.heatmap(data.corr(),annot=True,linewidth=0.6,fmt='.1f')#Heatmap to visualize the correlation between differ
```

Out[15]: <Axes: >



In [16]:
```
sns.scatterplot(data=data,x='exang',y='thalach',hue='target')#Scatter plot to visualize the relationship betwee
plt.show()
```



In [ ]: