In [1]:

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

In [2]:

```
df=pd.read_csv("C:\\Users\\tejas\\Downloads\\heart.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	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

In [4]:

```
df.tail()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
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]:

```
df.columns.values
```

Out[5]:

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

In [6]:

```
#checking for null values
df.isna().sum()
```

Out[6]:

```
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 [7]:

```
#concise summary of our dataset
df.info()
```

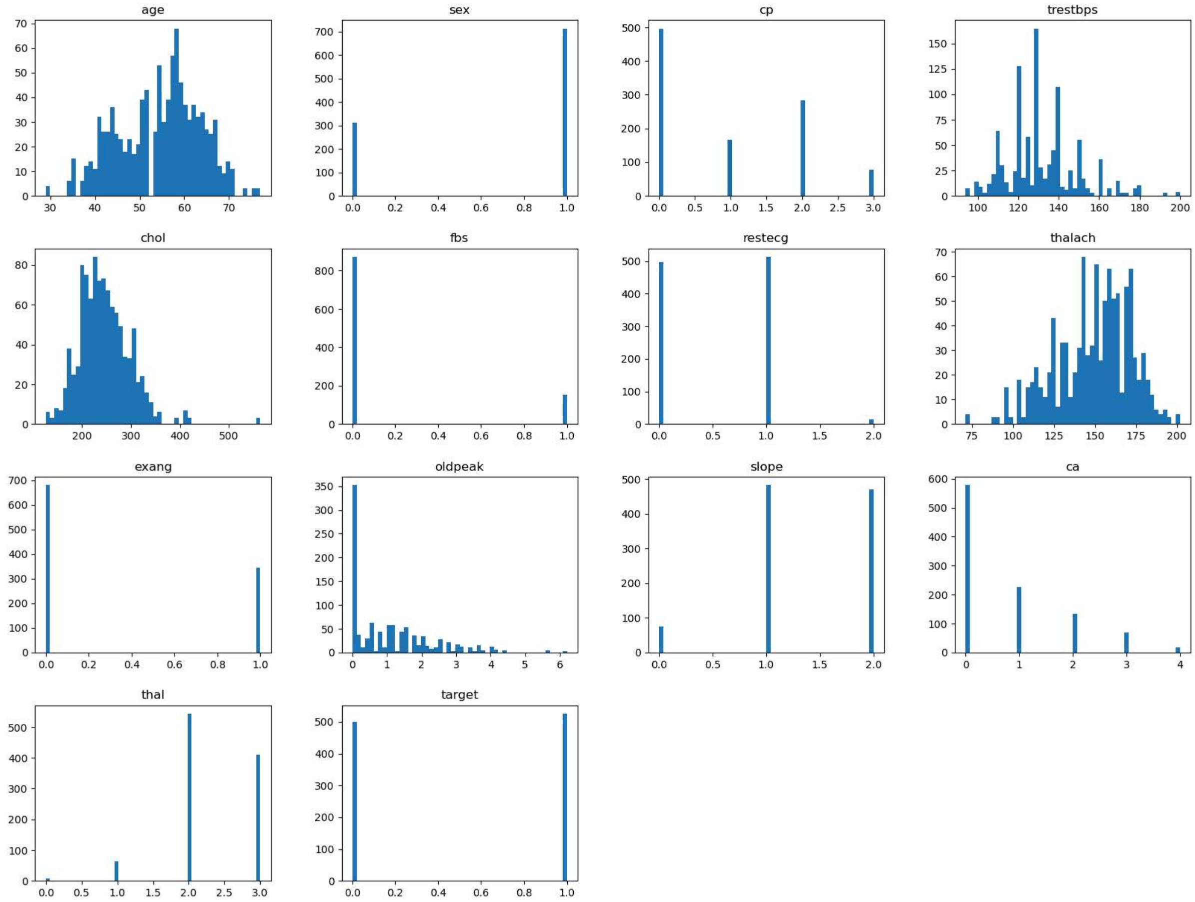
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 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
```

In [8]:

```
#plotting histograms of all numeric values
df.hist(bins=50,grid=False,figsize=(20,15))
```

Out[8]:

```
array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
        <Axes: title={'center': 'cp'}>,
        <Axes: title={'center': 'trestbps'}>],
       [<Axes: title={'center': 'chol'}>,
        <Axes: title={'center': 'fbs'}>,
        <Axes: title={'center': 'restecg'}>,
        <Axes: title={'center': 'thalach'}>],
       [<Axes: title={'center': 'exang'}>,
        <Axes: title={'center': 'oldpeak'}>,
        <Axes: title={'center': 'slope'}>,
        <Axes: title={'center': 'ca'}>],
       [<Axes: title={'center': 'thal'}>,
        <Axes: title={'center': 'target'}>, <Axes: >, <Axes: >]],
      dtype=object)
```



In [9]:

```
#Generating descriptive statistics
df.describe()
```

Out[9]:

	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

In [11]:

```
Questions=["1)How many people have heart disease and how many people dont have hear
"2)people of which sex has most heart disease?",
"3)people of of which sex has which type of chest pain most?",
"4)people with which chest pain are most prone to have heart disease?",
"5)What is the average age of individuals with heart disease compared to those with
"6) How does cholesterol level correlate with the presence of heart disease?",
"7)Is there a relationship between exercise induced angina (exang) and the severity
```

In [12]:

```
df.target.value_counts()
```

Out[12]:

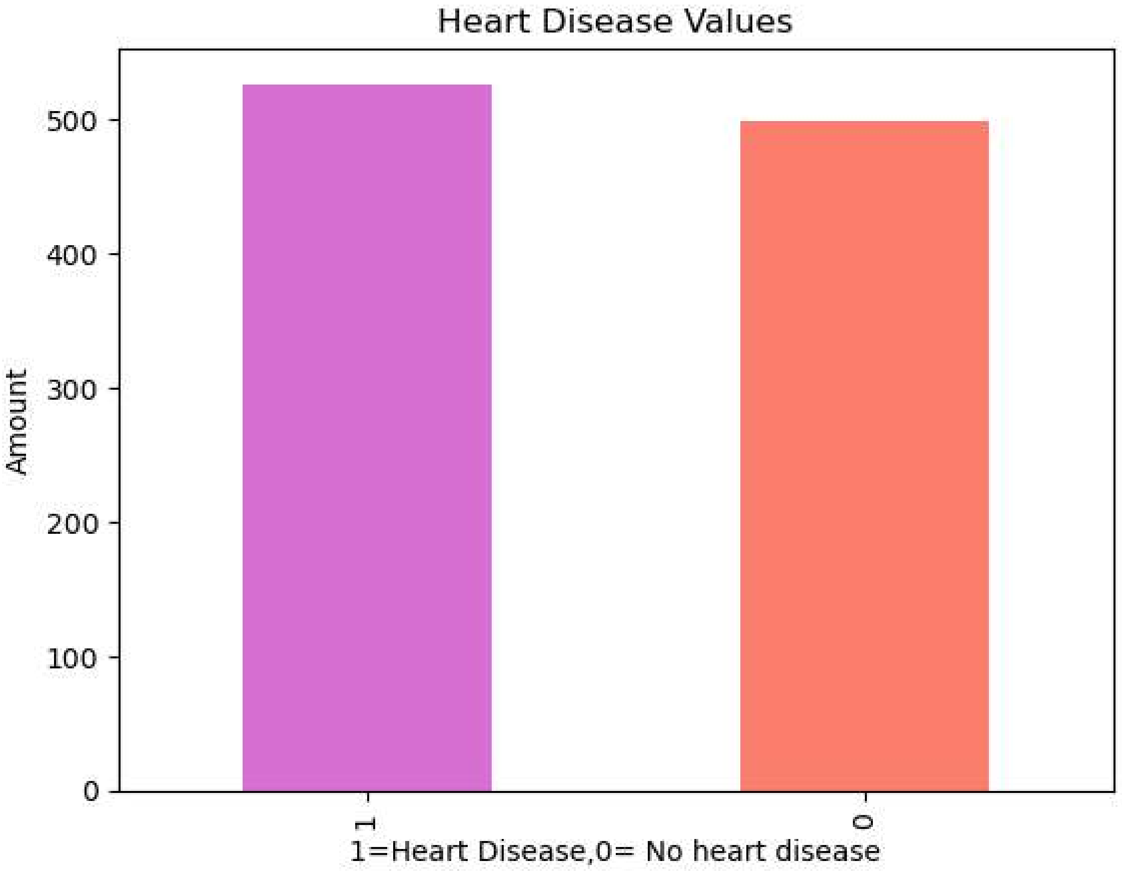
```
target
1    526
0    499
Name: count, dtype: int64
```

In [13]:

```
df.target.value_counts().plot(kind='bar',color=["orchid","salmon"])
plt.title("Heart Disease Values")
plt.xlabel("1=Heart Disease,0= No heart disease")
plt.ylabel("Amount")
```

Out[13]:

```
Text(0, 0.5, 'Amount')
```



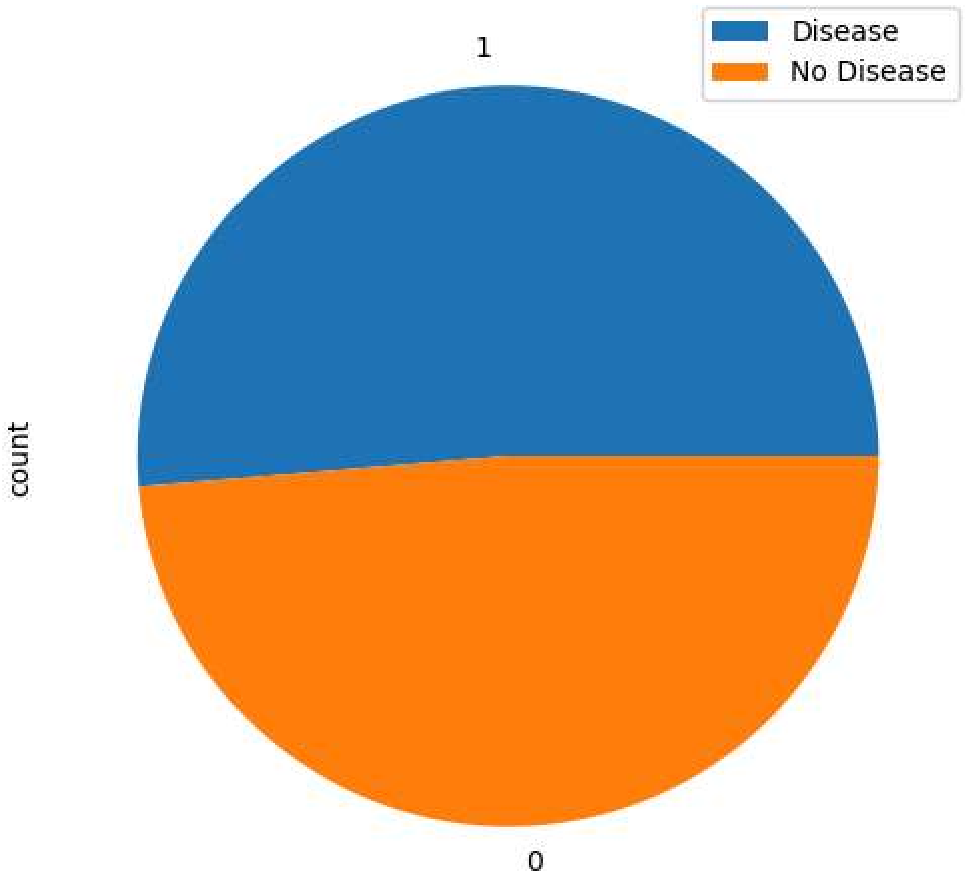
In [14]:

```
#plotting a pie chart

df.target.value_counts().plot(kind='pie',figsize=(8,6))
plt.legend(["Disease","No Disease"])
```

Out[14]:

```
<matplotlib.legend.Legend at 0x25308bc3d10>
```

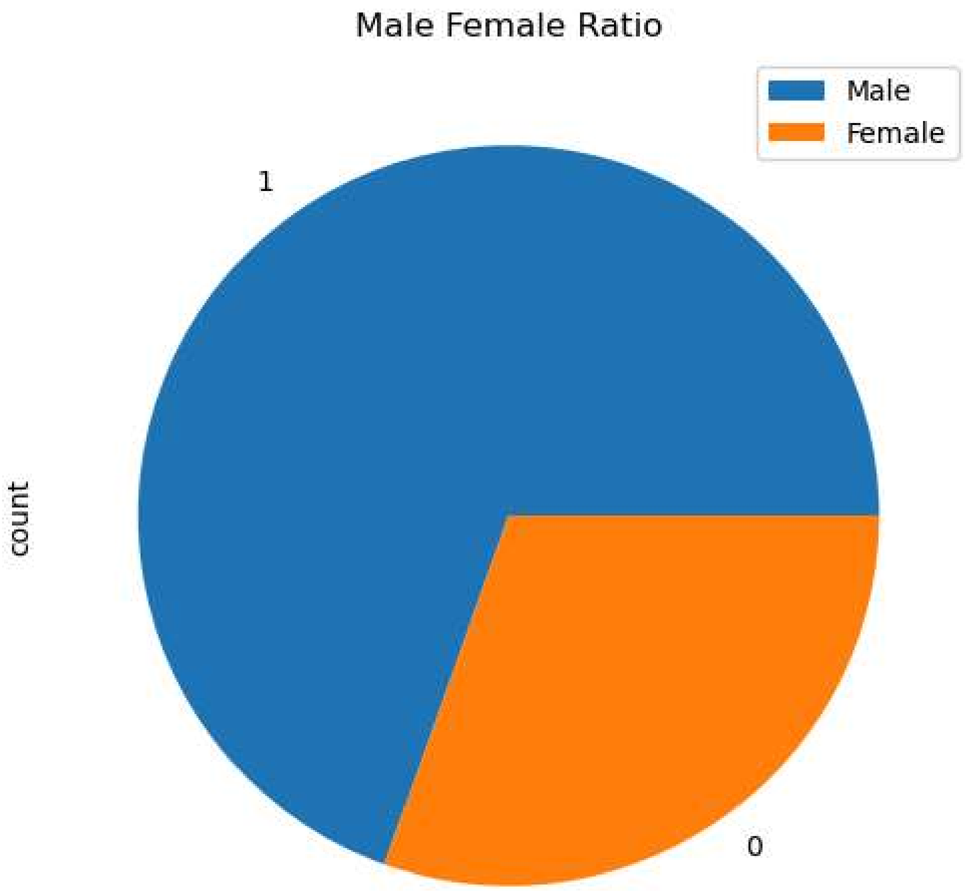


```
In [15]: #check how many male and female are in the dataset
         df.sex.value_counts()

Out[15]: sex
         1    713
         0    312
         Name: count, dtype: int64

In [16]: df.sex.value_counts().plot(kind='pie',figsize=(8,6))
         plt.title('Male Female Ratio')
         plt.legend(["Male","Female"])

Out[16]: <matplotlib.legend.Legend at 0x2530b3a0b90>
```



In [24]:

```
#Q2

pd.crosstab(df.target,df.sex)
```

Out[24]:

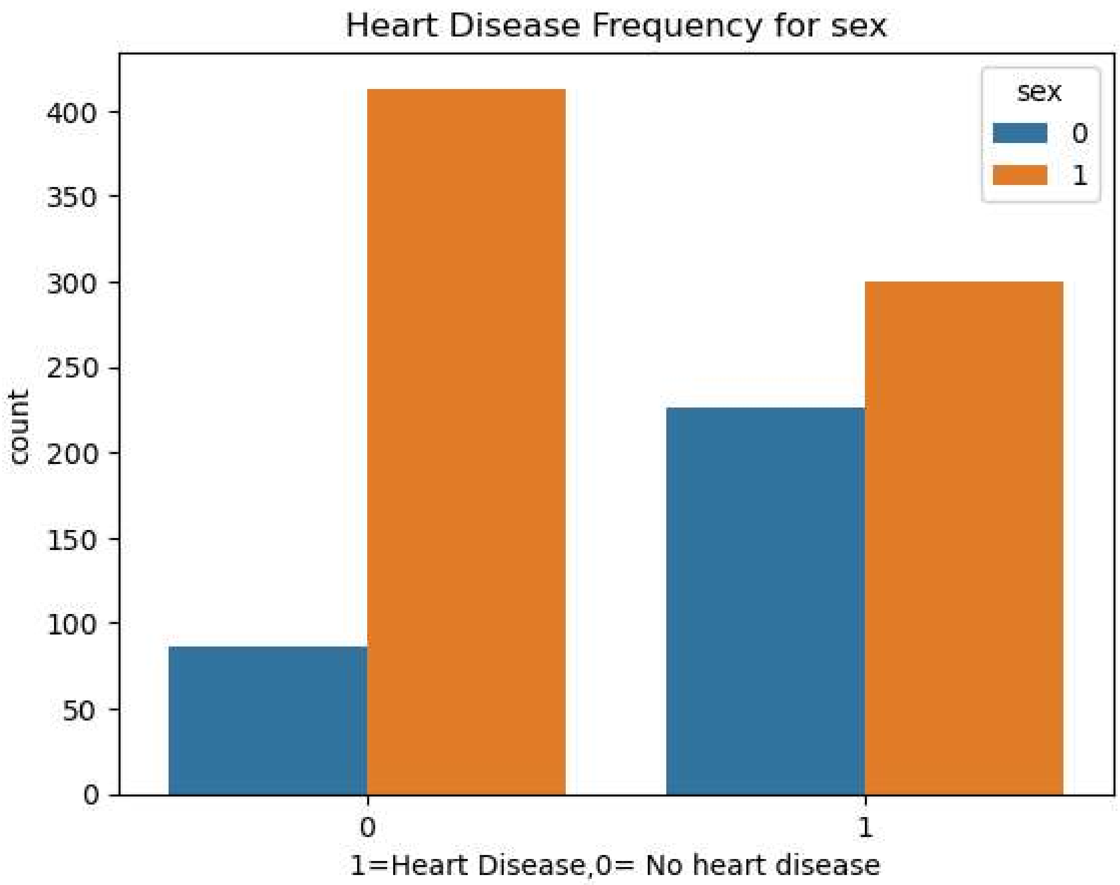
sex	0	1
target		
0	86	413
1	226	300

In [25]:

```
sns.countplot(x='target',data=df,hue='sex')
plt.title("Heart Disease Frequency for sex")
plt.xlabel("1=Heart Disease,0= No heart disease")
```

Out[25]:

```
Text(0.5, 0, '1=Heart Disease,0= No heart disease')
```



In [26]:

```
#Q3

df.cp.value_counts()
```

Out[26]:

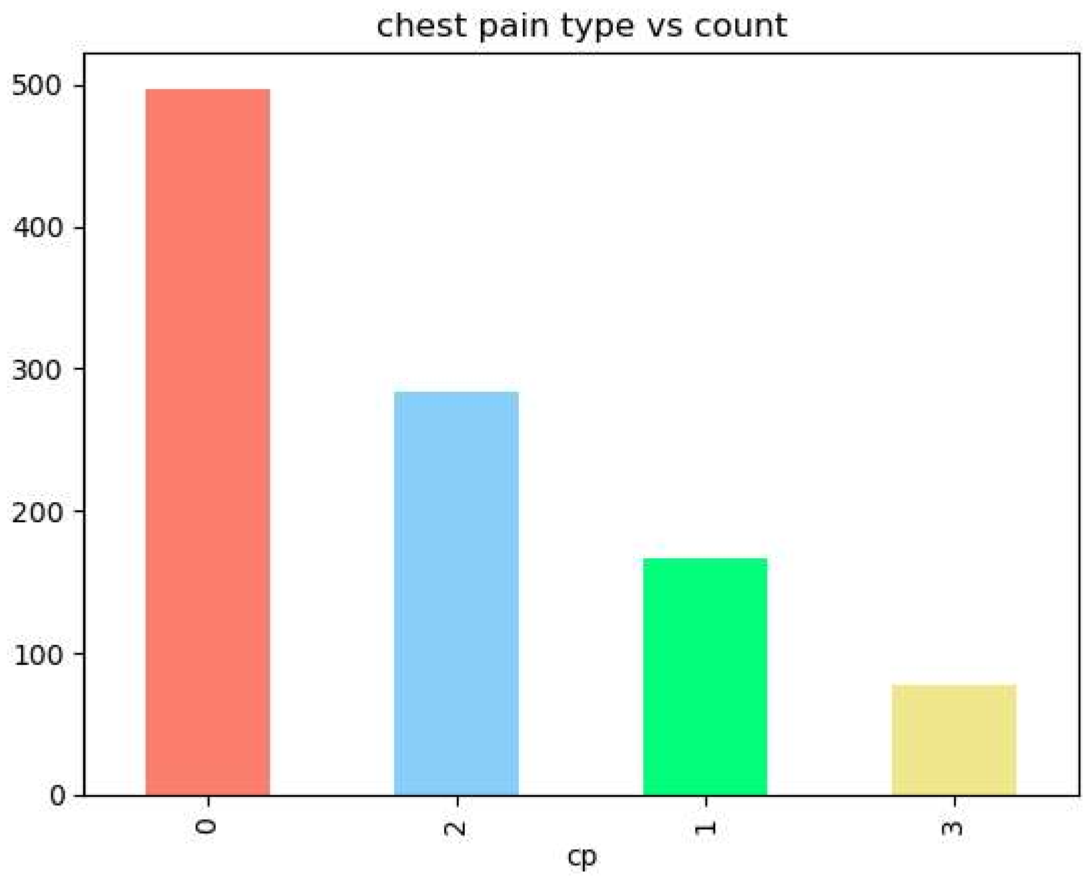
```
cp
0    497
2    284
1    167
3     77
Name: count, dtype: int64
```

In [28]:

```
df.cp.value_counts().plot(kind='bar',color=['salmon','lightskyblue','springgreen','
plt.title("chest pain type vs count")
```

Out[28]:

```
Text(0.5, 1.0, 'chest pain type vs count')
```

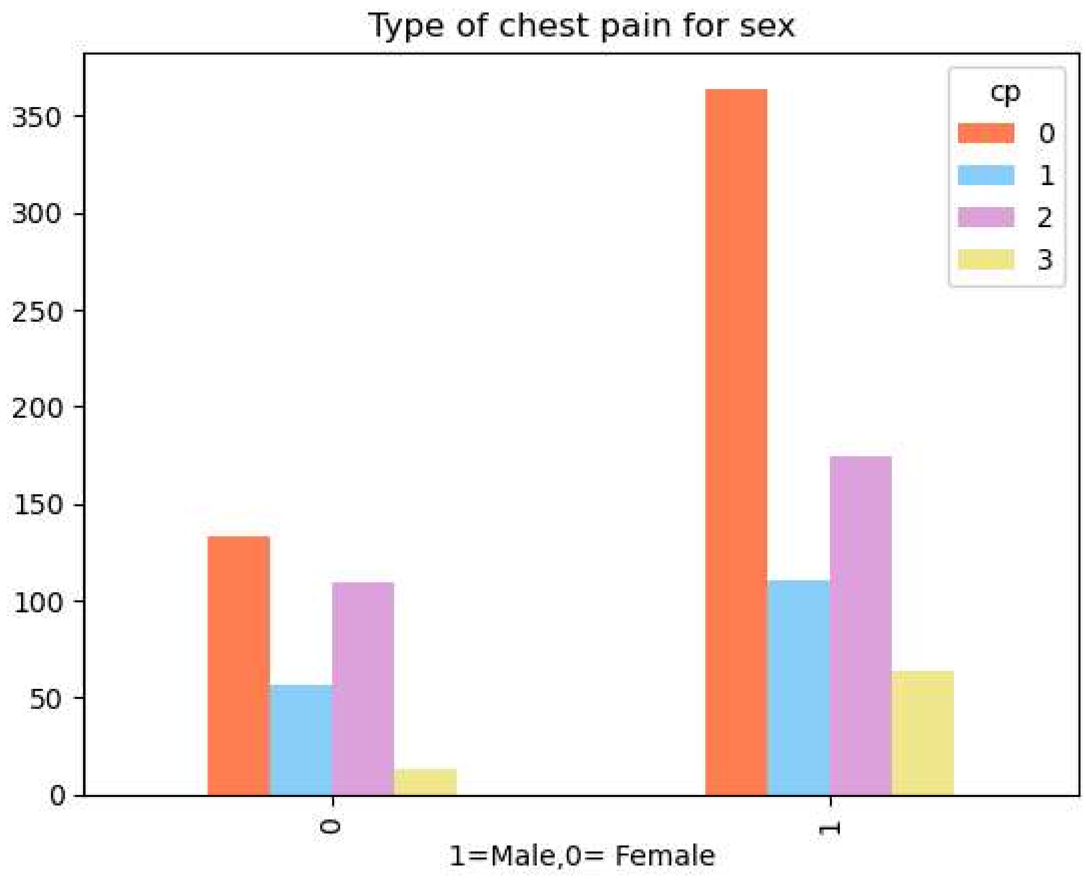


```
In [29]: pd.crosstab(df.sex,df.cp)
```

```
Out[29]:  cp     0    1    2   3
         sex
           0   133   57  109  13
           1   364  110  175  64
```

```
In [30]: pd.crosstab(df.sex,df.cp).plot(kind='bar',color=['coral','lightskyblue','plum','kha
         plt.title("Type of chest pain for sex")
         plt.xlabel("1=Male,0= Female")
```

```
Out[30]: Text(0.5, 0, '1=Male,0= Female')
```



In [31]:

```
#Q4
pd.crosstab(df.cp,df.target)
```

Out[31]:

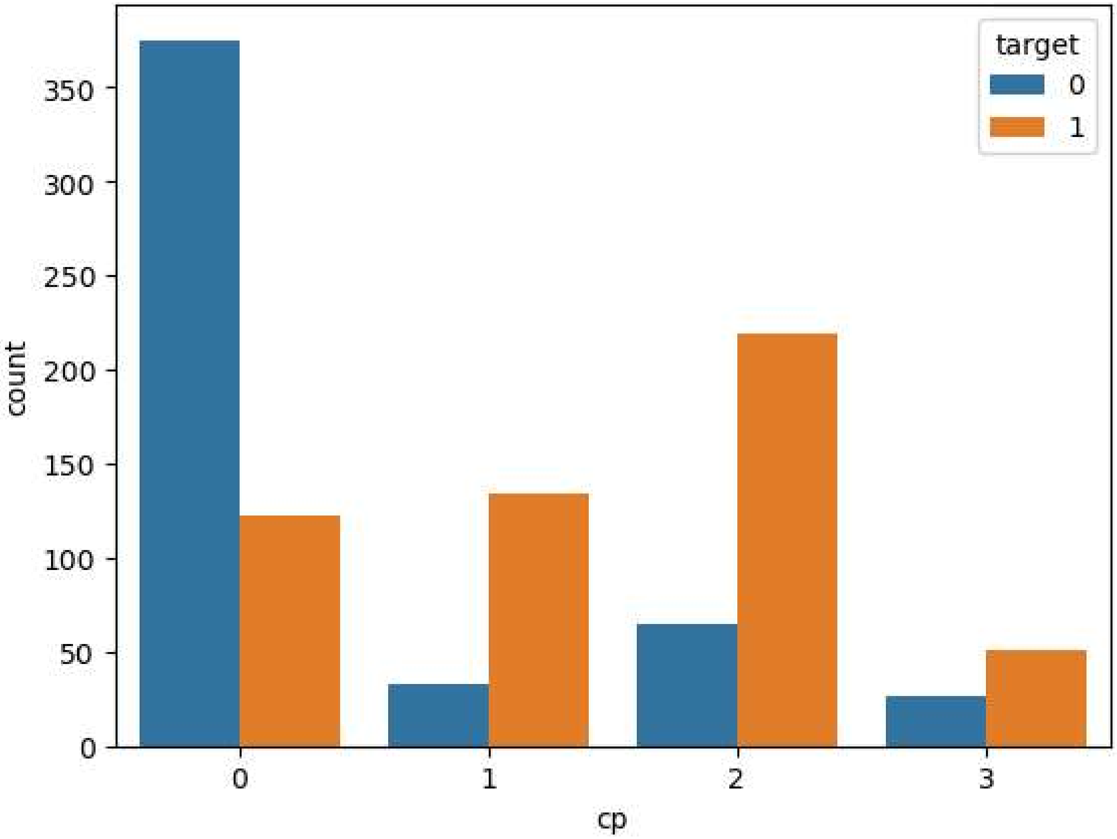
target	0	1
cp		
0	375	122
1	33	134
2	65	219
3	26	51

In [32]:

```
sns.countplot(x='cp',data=df,hue='target')
```

Out[32]:

```
<Axes: xlabel='cp', ylabel='count'>
```



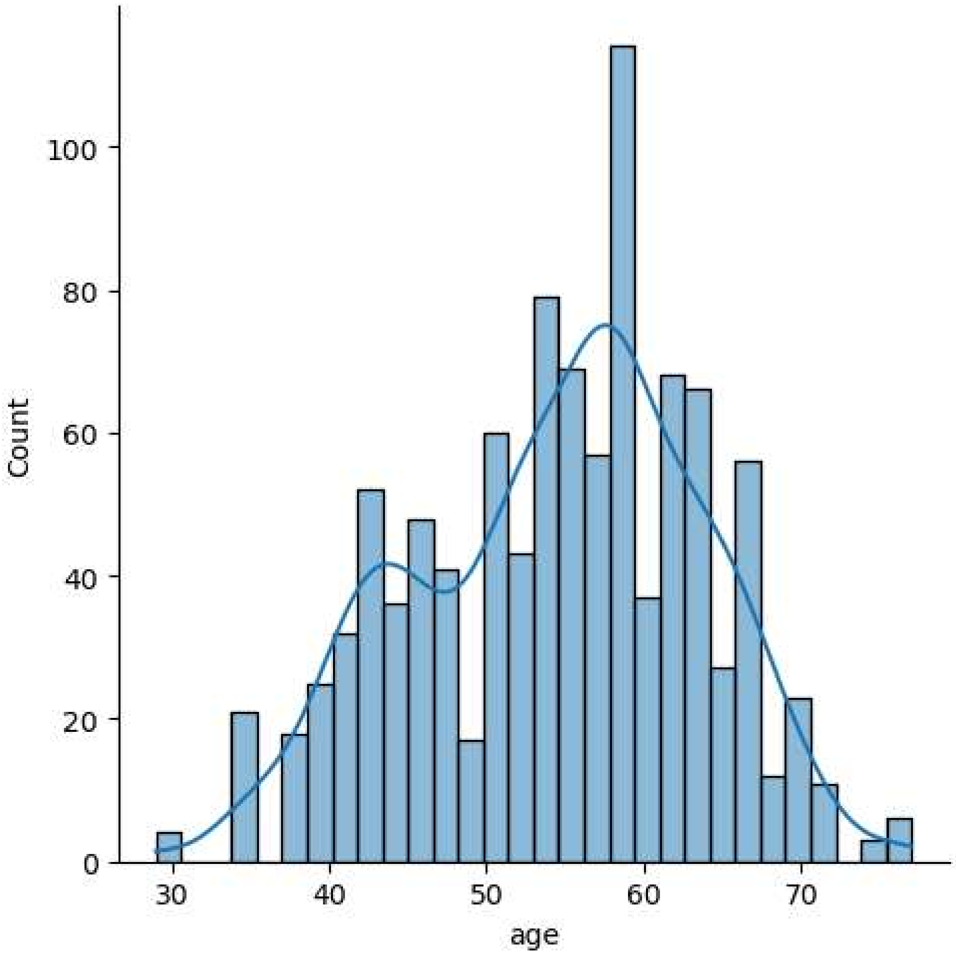
In [33]:

```
#create a distribution plot with normal distribution curve
sns.displot(x='age',data=df,bins=30,kde=True)
```

```
C:\Users\tejas\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: T
he figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[33]:

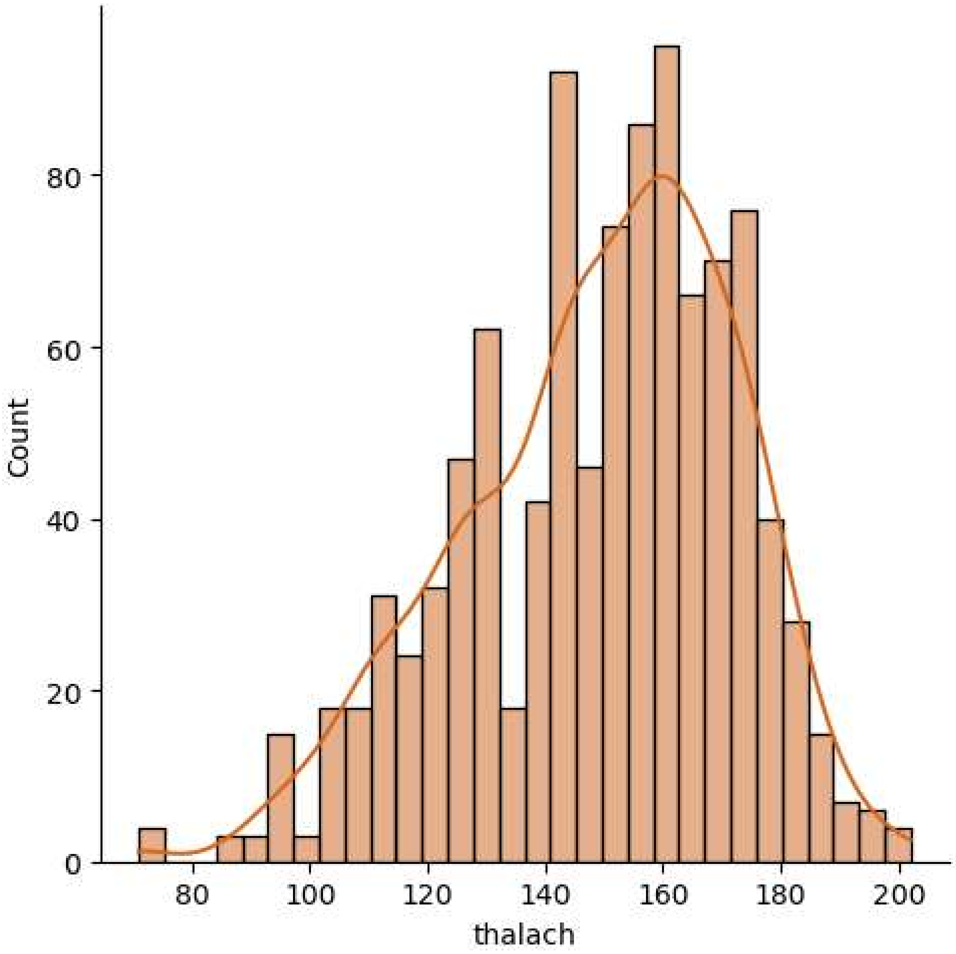
```
<seaborn.axisgrid.FacetGrid at 0x2530d05ef90>
```



In [37]:
```
#max heart rate distribution plot
sns.displot(x='thalach',data=df,bins=30,kde=True,color='chocolate')
```

```
C:\Users\tejas\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: T
he figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[37]:
```
<seaborn.axisgrid.FacetGrid at 0x2530d07d190>
```



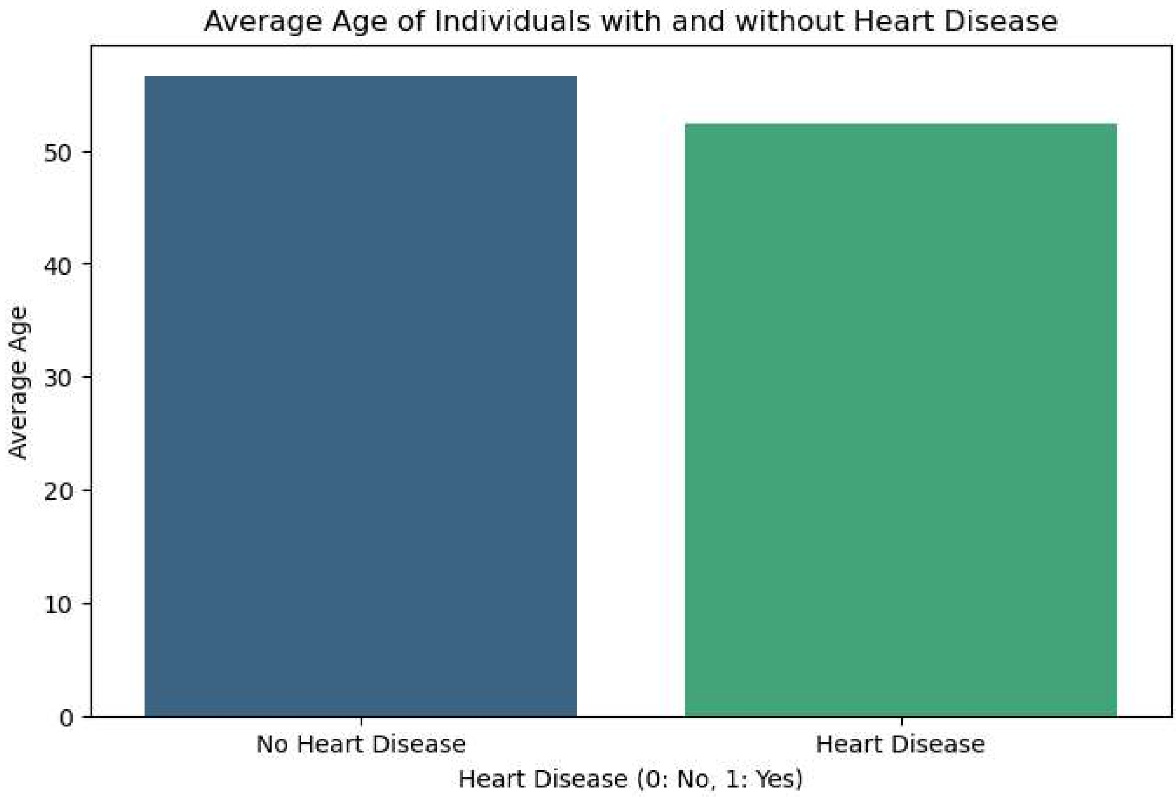
In [40]:

```
#Q5
# Average age of individuals with and without heart disease
avg_age = df.groupby('target')['age'].mean()
print("Average age of individuals with and without heart disease:")
print(avg_age)
```

```
Average age of individuals with and without heart disease:
target
0    56.569138
1    52.408745
Name: age, dtype: float64
```

In [41]:

```
# Bar chart for average age
plt.figure(figsize=(8, 5))
sns.barplot(x=avg_age.index, y=avg_age.values, palette='viridis')
plt.title('Average Age of Individuals with and without Heart Disease')
plt.xlabel('Heart Disease (0: No, 1: Yes)')
plt.ylabel('Average Age')
plt.xticks(ticks=[0, 1], labels=['No Heart Disease', 'Heart Disease'])
plt.show()
```



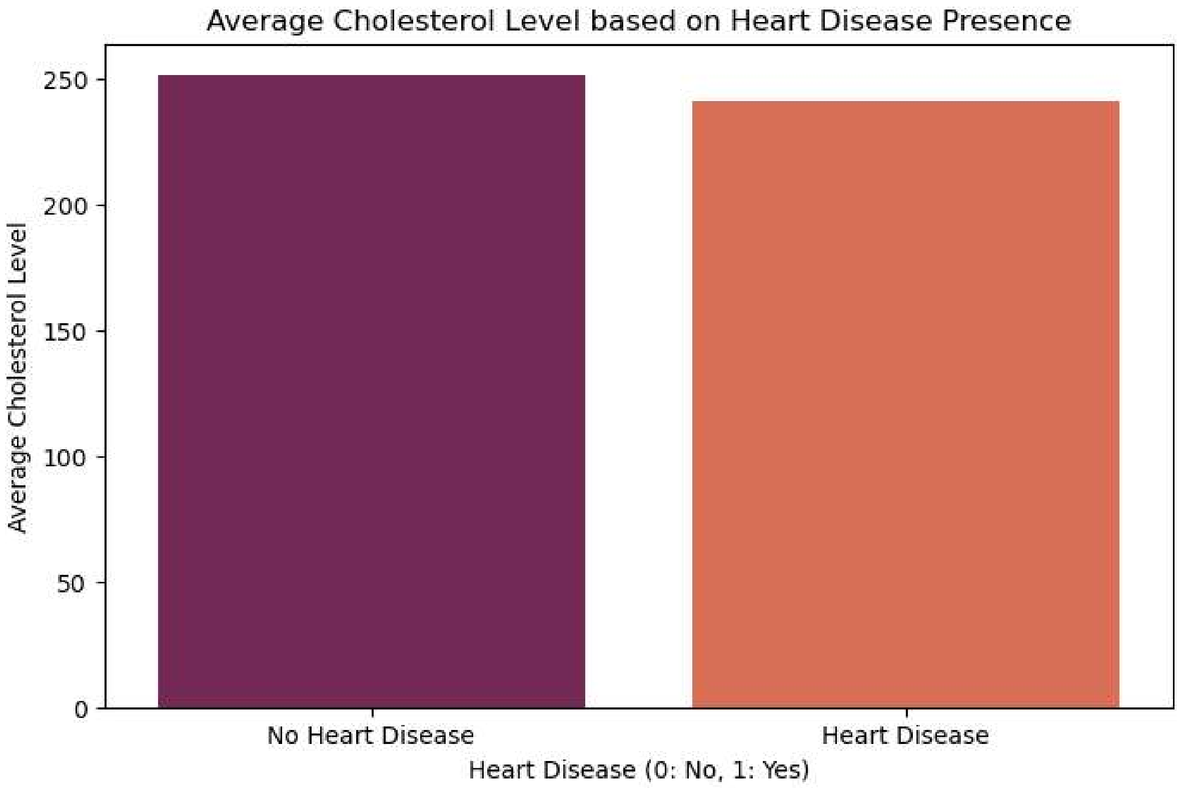
In [42]:

```
# Q6. Correlation between cholesterol level and heart disease
cholesterol_corr = df[['chol', 'target']].groupby('target').mean()
print("\nAverage Cholesterol Level based on Heart Disease presence:")
print(cholesterol_corr)
```

```
Average Cholesterol Level based on Heart Disease presence:
              chol
target
0       251.292585
1       240.979087
```

In [43]:

```
# Bar chart for average cholesterol
plt.figure(figsize=(8, 5))
sns.barplot(x=cholesterol_corr.index, y=cholesterol_corr['chol'], palette='rocket')
plt.title('Average Cholesterol Level based on Heart Disease Presence')
plt.xlabel('Heart Disease (0: No, 1: Yes)')
plt.ylabel('Average Cholesterol Level')
plt.xticks(ticks=[0, 1], labels=['No Heart Disease', 'Heart Disease'])
plt.show()
```



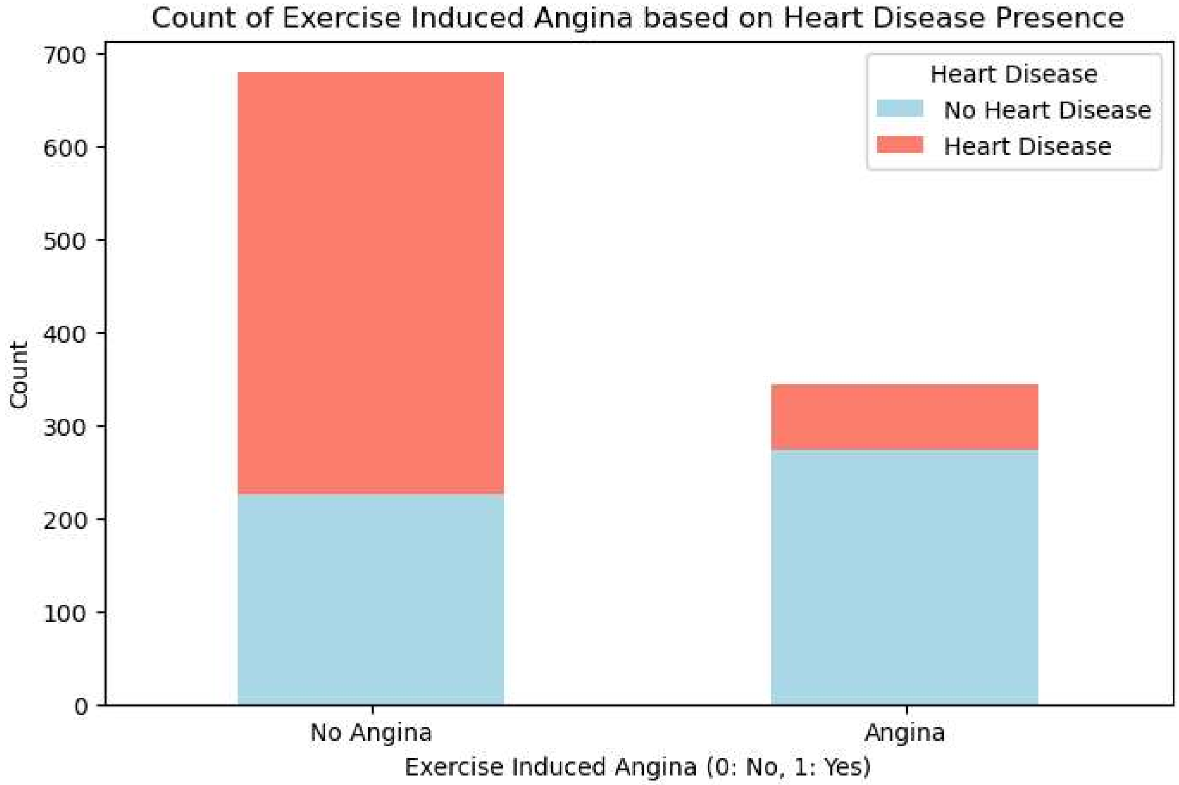
In [44]:

```
# Q7. Relationship between exercise induced angina and heart disease
exang_count = df.groupby(['exang', 'target']).size().unstack(fill_value=0)
print("\nCount of Exercise Induced Angina based on Heart Disease presence:")
print(exang_count)
```

```
Count of Exercise Induced Angina based on Heart Disease presence:
target    0    1
exang
0       225  455
1       274   71
```

In [45]:

```
# Bar chart for exercise induced angina
exang_count.plot(kind='bar', stacked=True, figsize=(8, 5), color=['lightblue', 'sal
plt.title('Count of Exercise Induced Angina based on Heart Disease Presence')
plt.xlabel('Exercise Induced Angina (0: No, 1: Yes)')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['No Angina', 'Angina'], rotation=0)
plt.legend(title='Heart Disease', labels=['No Heart Disease', 'Heart Disease'])
plt.show()
```


Count of Exercise Induced Angina based on Heart Disease Presence
Heart Disease
No Heart Disease
Heart Disease
Count
700
600
500
400
300
200
100
0
No Angina
Angina
Exercise Induced Angina (0: No, 1: Yes)

In [ ]: