In [58]:
```
# Importing Libraries

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

In [59]:
```
# Importing the Dataset
```

In [60]:

Out[60]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
15	34	0	1	118	210	0	1	192	0	0.7	2	0	2	
80	50	1	2	140	233	0	1	163	0	0.6	1	1	3	
211	67	1	0	100	299	0	0	125	1	0.9	1	2	2	
29	55	0	0	180	327	0	2	117	1	3.4	1	0	2	
676	60	1	0	130	253	0	1	144	1	1.4	2	1	3	
708	60	0	2	120	178	1	1	96	0	0.0	2	0	2	
788	62	0	0	138	294	1	1	106	0	1.9	1	3	2	
493	55	1	0	132	353	0	1	132	1	1.2	1	1	3	
88	62	0	0	140	268	0	0	160	0	3.6	0	2	2	
551	54	1	0	122	286	0	0	116	1	3.2	1	2	2	

In [61]:

```
<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 [62]:
```
# Checking Duplicates in the Dataset
```

Out[62]: 723

In [63]:
```
# Removing the Duplicate datapoints
```

In [64]:
```

```

Out[64]: (302, 14)

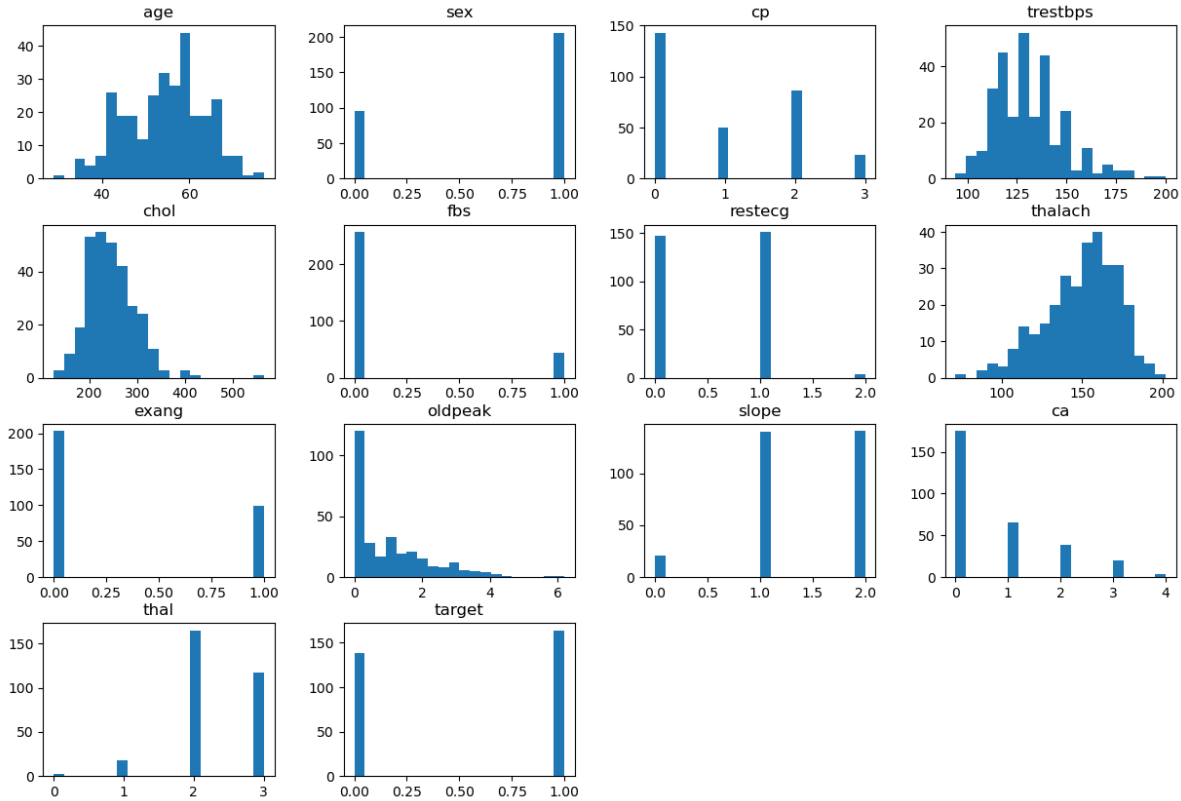
In [65]:
```

```

Out[65]:

	age	sex	cp	trestbps	chol	fbs	restecg	th
count	302.00000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.0
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.5
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.9
min	29.00000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.0
25%	48.00000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.2
50%	55.50000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.5
75%	61.00000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.0
max	77.00000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.0

In [66]:
```
# Checking the Distribution of the features
```



In [67]:
```
# understanding the correalation between the features
plt.figure(figsize=(15,5))
sns.heatmap(df.corr(),annot=True,linewidths=1)
```

In [68]:
```
# Converting Target from  Numarical Data to Catogorical data

def convert(row):
    if row==1:
        return 'Heart_Disease'
    elif row==0:
```

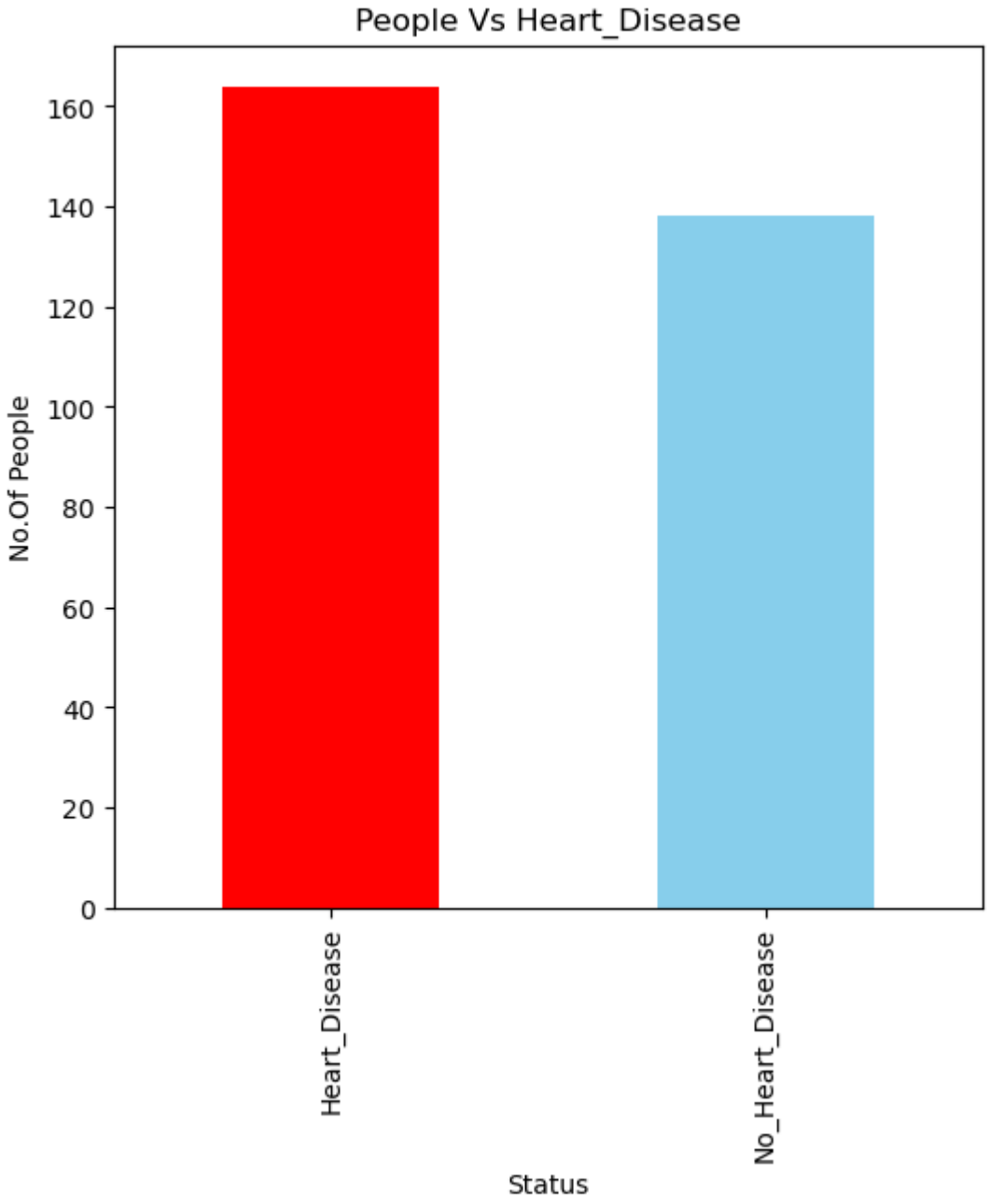
In [69]:
```
df['target1']=df['target'].apply(convert)
```

Out[69]:

	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

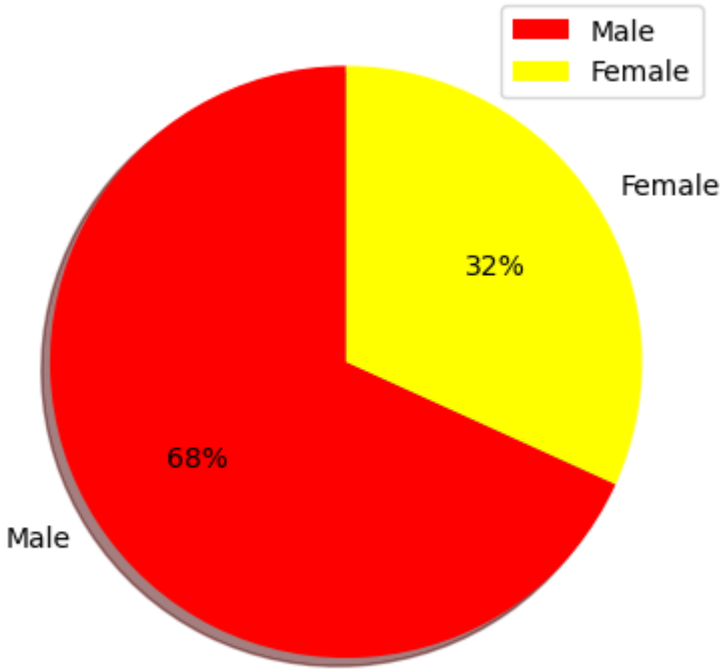
In [70]:
```
# Checking How many people Having Heart Dicease

plt.figure(figsize=(6,6))
df.target1.value_counts().plot(kind='bar',color=['red','skyblue'])
plt.ylabel('No.Of People')
plt.xlabel('Status')
plt.title('People Vs Heart_Disease')
```



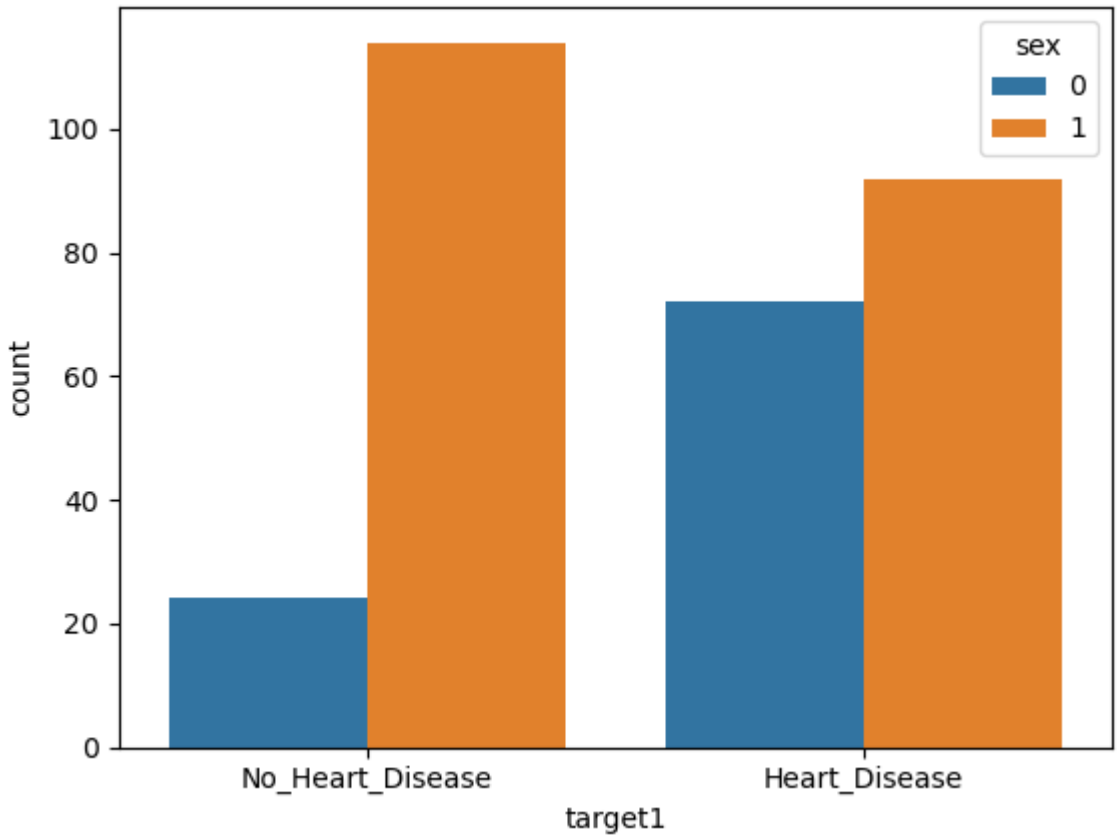
In [71]:

In [72]:
```
plt.pie(x,colors=['red','yellow'],startangle=90,shadow=True,autopct='%.f%%'
plt.legend()
```



In [73]:

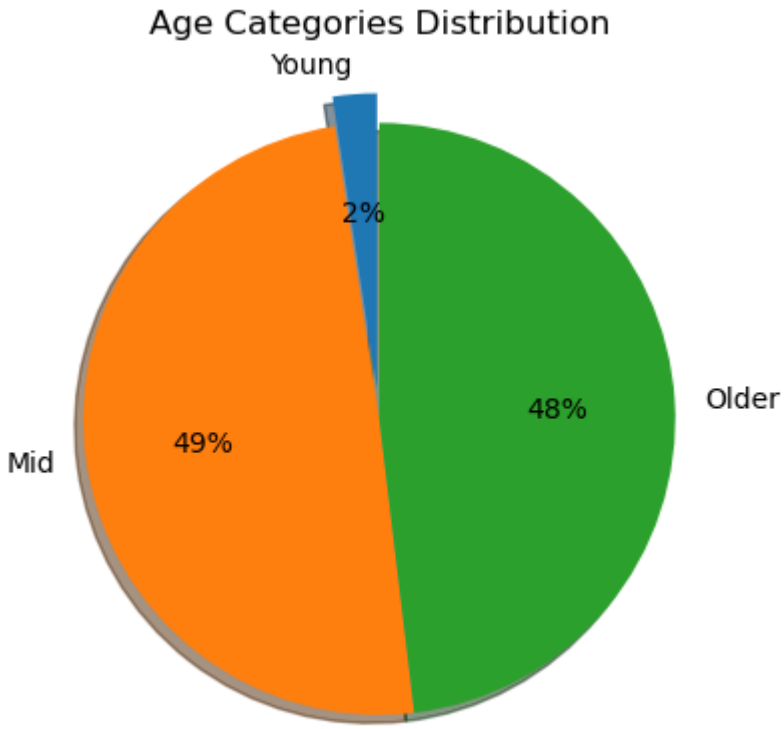
Out[73]: <Axes: xlabel='target1', ylabel='count'>



In [74]:
```
Young = df[((df['age']>=20) & (df['age']<=35))]
Mid = df[((df['age']>=36) & (df['age']<=55))]
```

In [75]:
```
l1=len(Young)
l2=len(Mid)
```

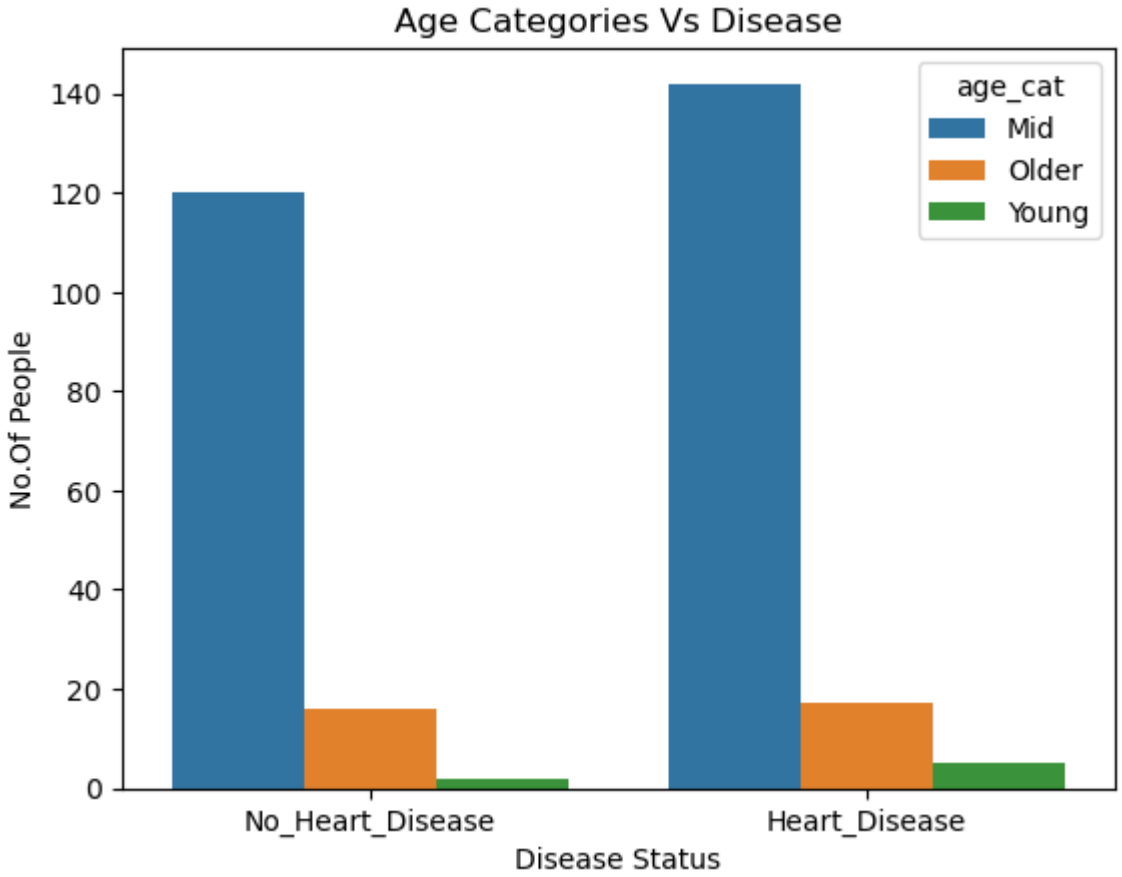
In [76]:
```
plt.pie(x=[l1,l2,l3],autopct='%.f%%',labels=['Young','Mid','Older'],explode
plt.title('Age Categories Distribution')
```



In [77]:
```
def age(row):
    if row<=35:
        return 'Young'
    elif row>65:
        return 'Older'
    else:
```

In [78]:

In [79]:
```
sns.countplot(data=df,x='target1',hue='age_cat')
plt.title('Age Categories Vs Disease')
plt.xlabel('Disease Status')
plt.ylabel('No.Of People')
```

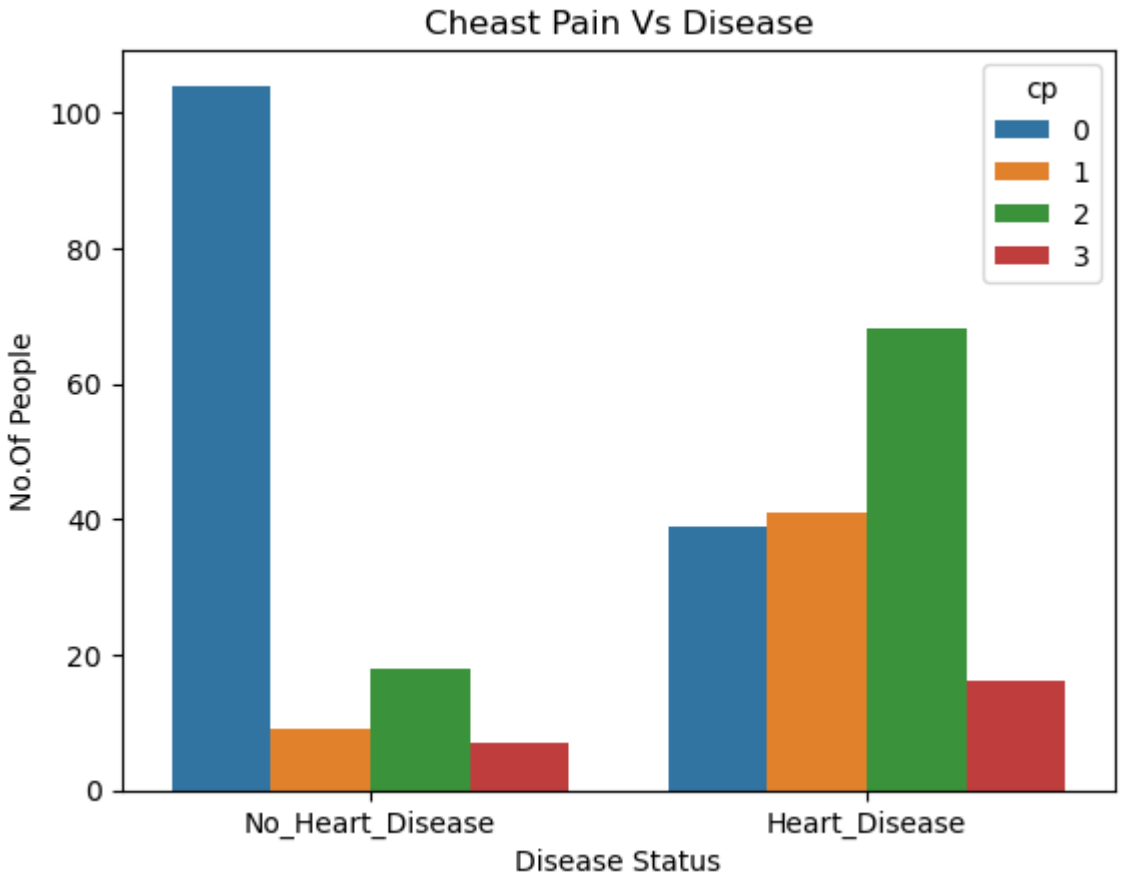


In [80]:

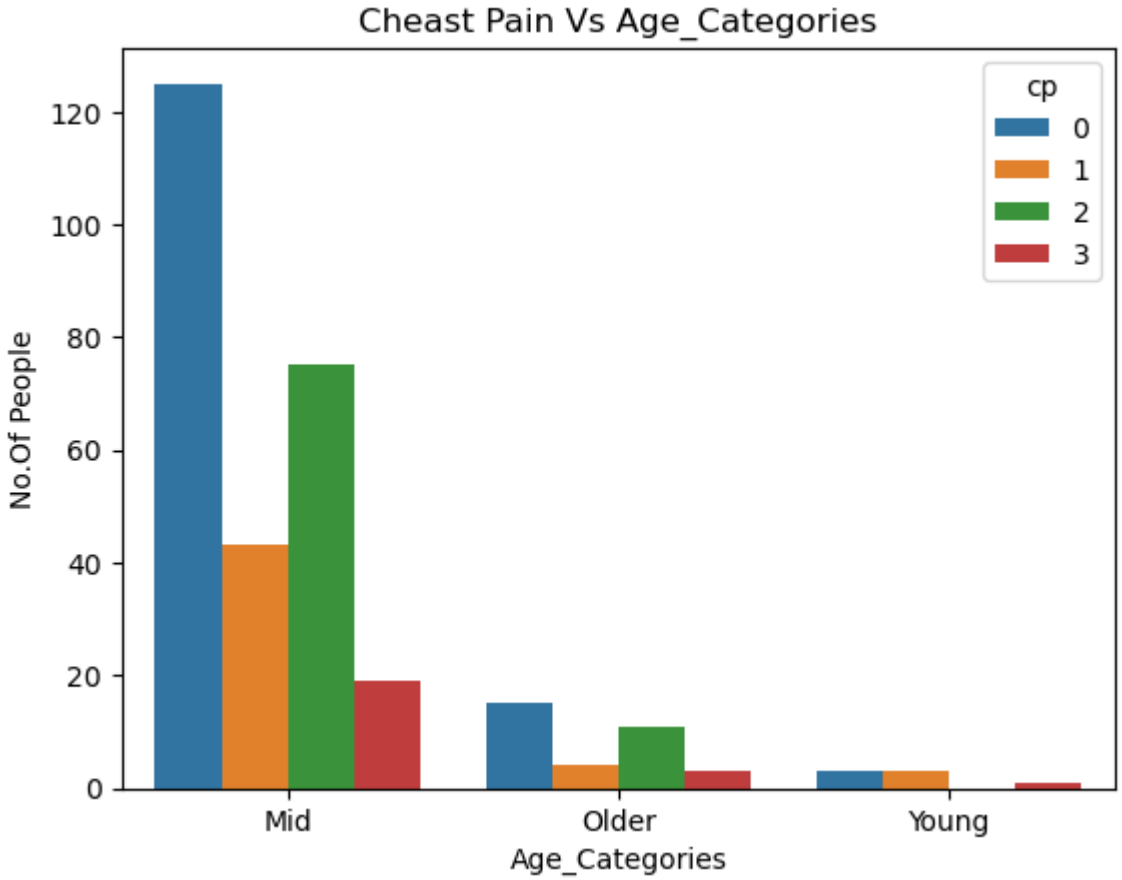
Out[80]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
12	34	0	1	118	210	0	1	192	0	0.7	2	0	2	
60	29	1	1	130	204	0	0	202	0	0.0	2	0	2	
143	34	1	3	118	182	0	0	174	0	0.0	2	0	2	
256	35	0	0	138	183	0	1	182	0	1.4	2	0	2	
281	35	1	1	122	192	0	1	174	0	0.0	2	0	2	

In [81]:
```
sns.countplot(data=df,x='target1',hue='cp')
plt.title('Cheast Pain Vs Disease')
plt.xlabel('Disease Status')
plt.ylabel('No.Of People')
```



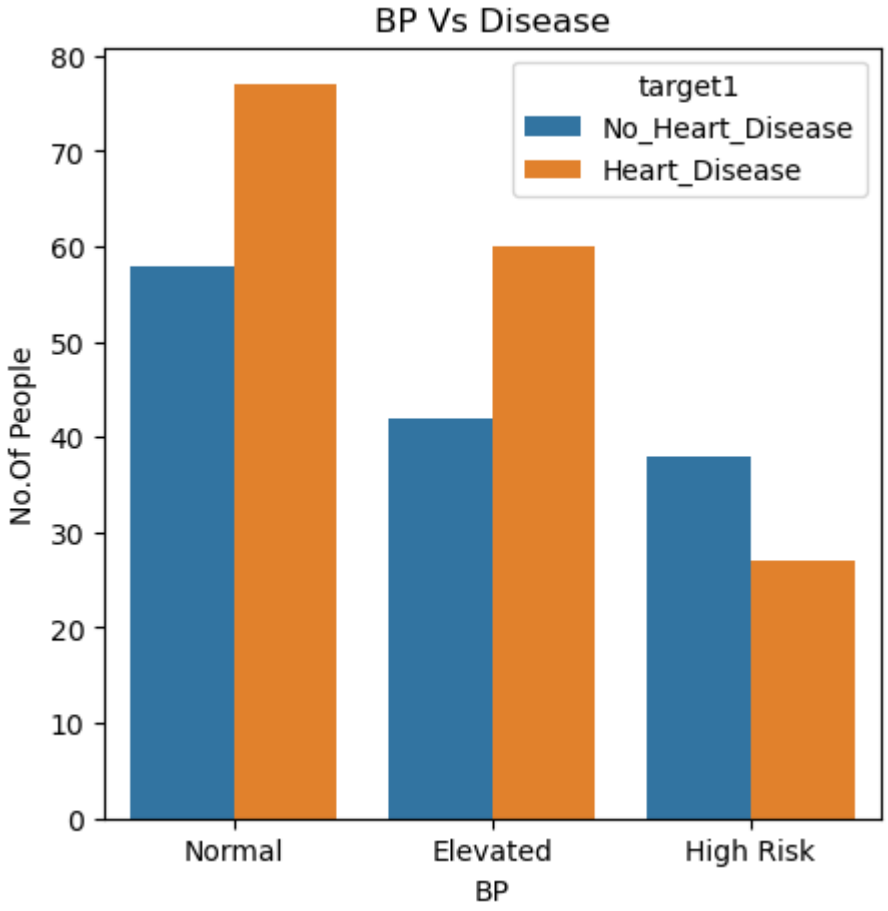
In [82]:
```
sns.countplot(data=df,x='age_cat',hue='cp')
plt.title('Cheast Pain Vs Age_Categories')
plt.xlabel('Age_Categories')
plt.ylabel('No.Of People')
```



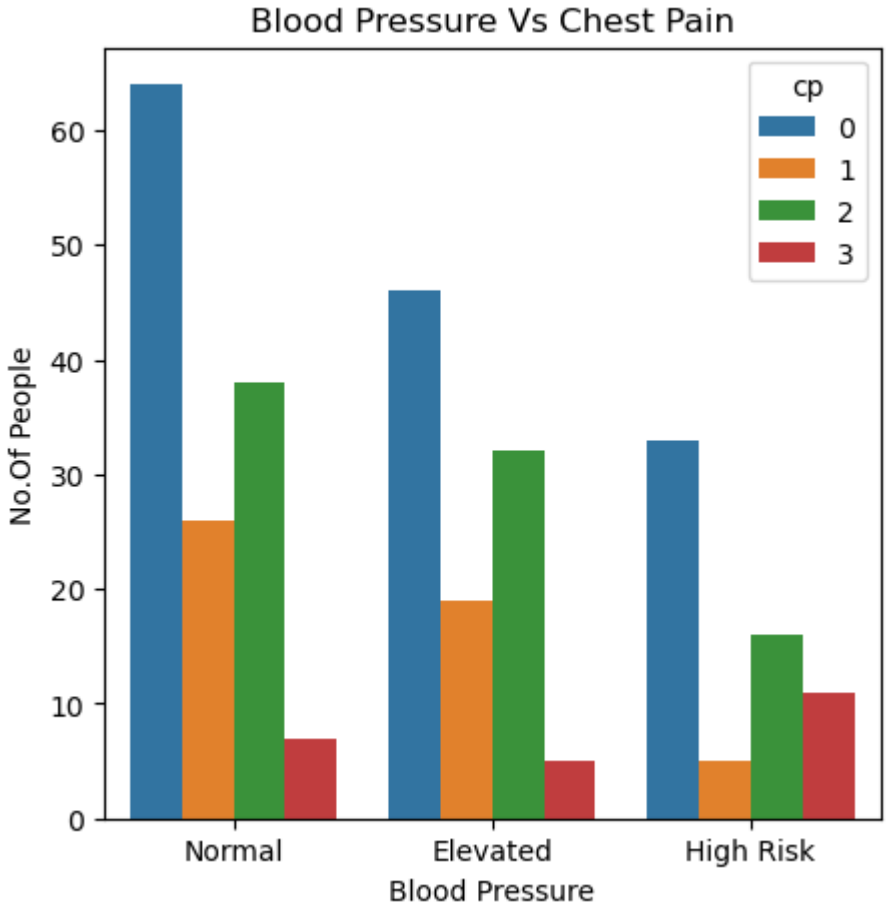
In [83]:
```
def bp(row):
    if row<=129:
        return 'Normal'
    elif row>140:
        return 'High Risk'
    else:
        return 'Elevated'
```

In [84]:

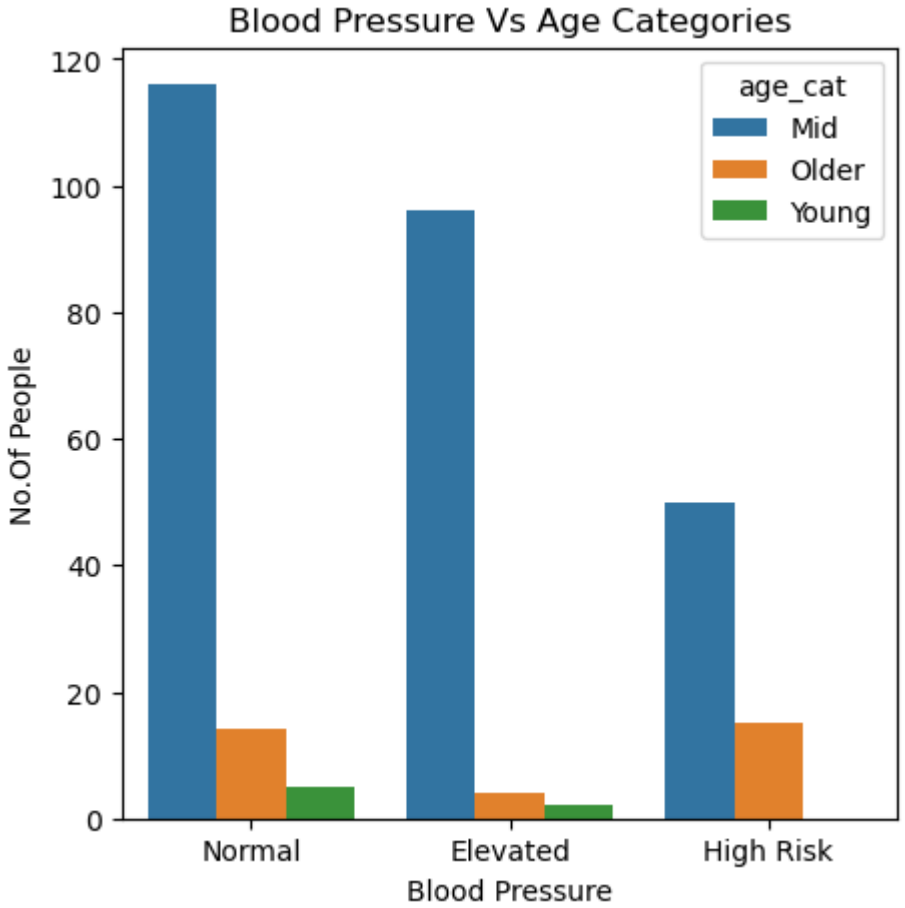
In [85]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='bp',hue='target1')
plt.title('BP Vs Disease')
plt.xlabel('BP')
plt.ylabel('No.Of People')
```



In [86]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='bp',hue='cp')
plt.title('Blood Pressure Vs Chest Pain')
plt.xlabel('Blood Pressure')
plt.ylabel('No.Of People')
```



In [87]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='bp',hue='age_cat')
plt.title('Blood Pressure Vs Age Categories')
plt.xlabel('Blood Pressure')
plt.ylabel('No.Of People')
```



In [88]:
```
def chol(row):
    if row<=150:
        return 'Normal'
    elif row>200:
        return 'High'
    else:
```

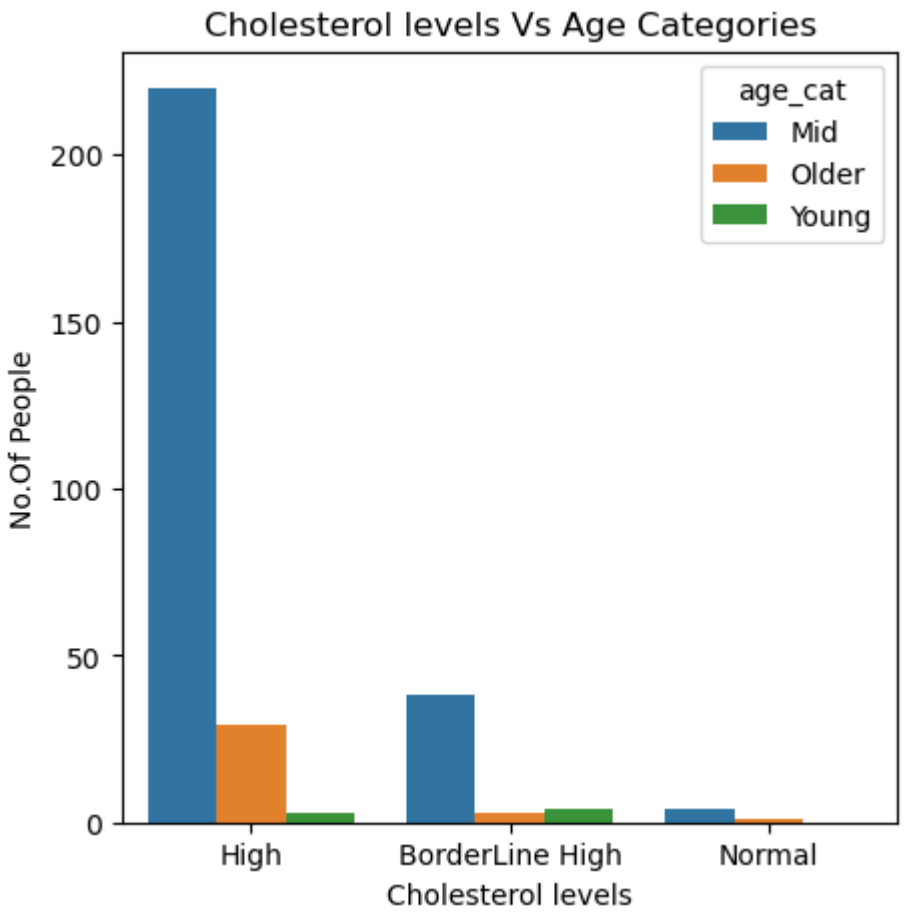
In [89]:

In [90]:

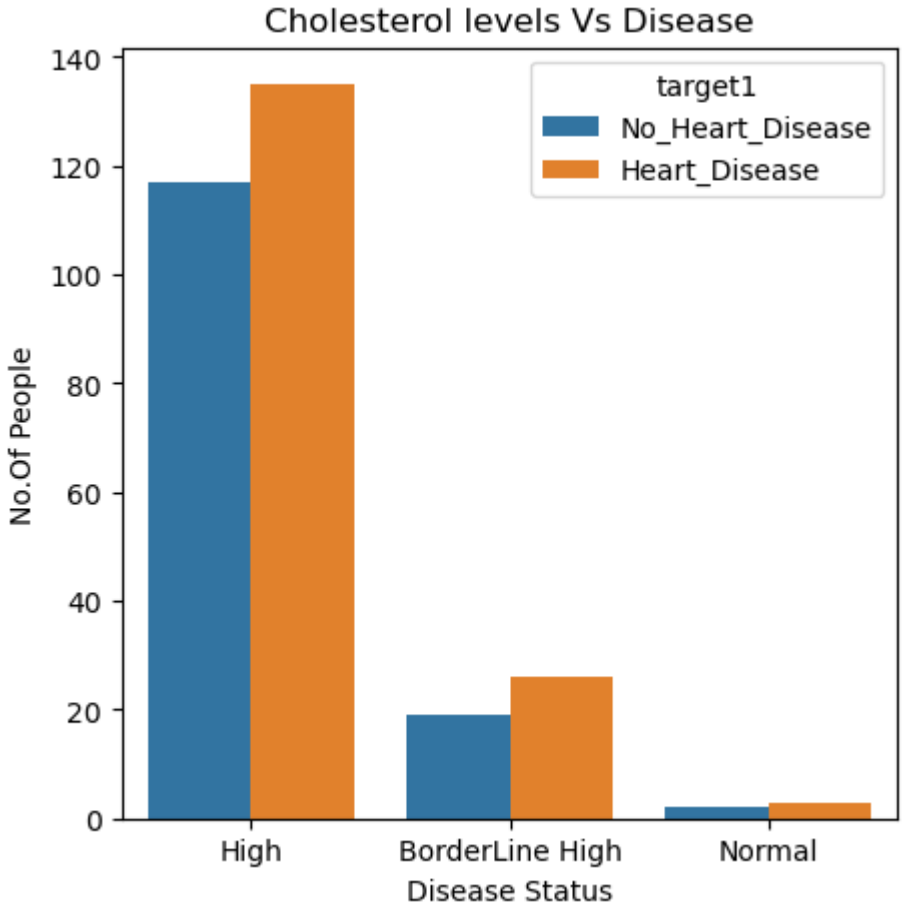
Out[90]:

	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

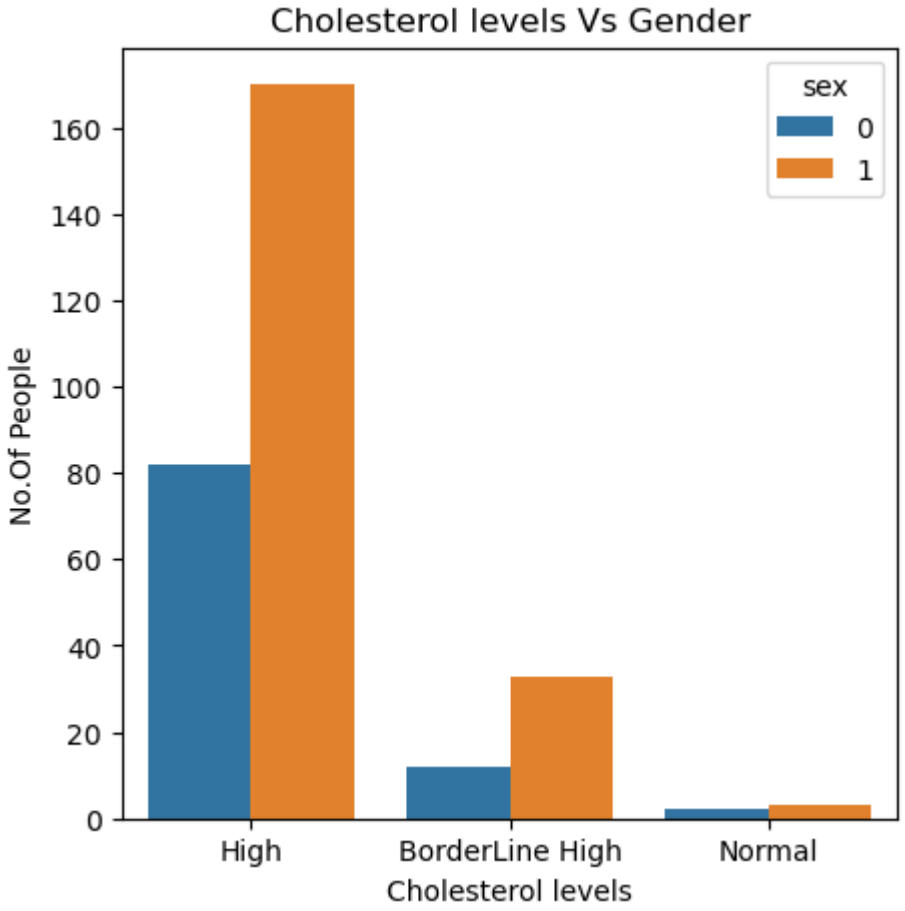
In [91]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='chol_cc',hue='age_cat')
plt.title('Cholesterol levels Vs Age Categories')
plt.xlabel('Cholesterol levels')
plt.ylabel('No.Of People')
```



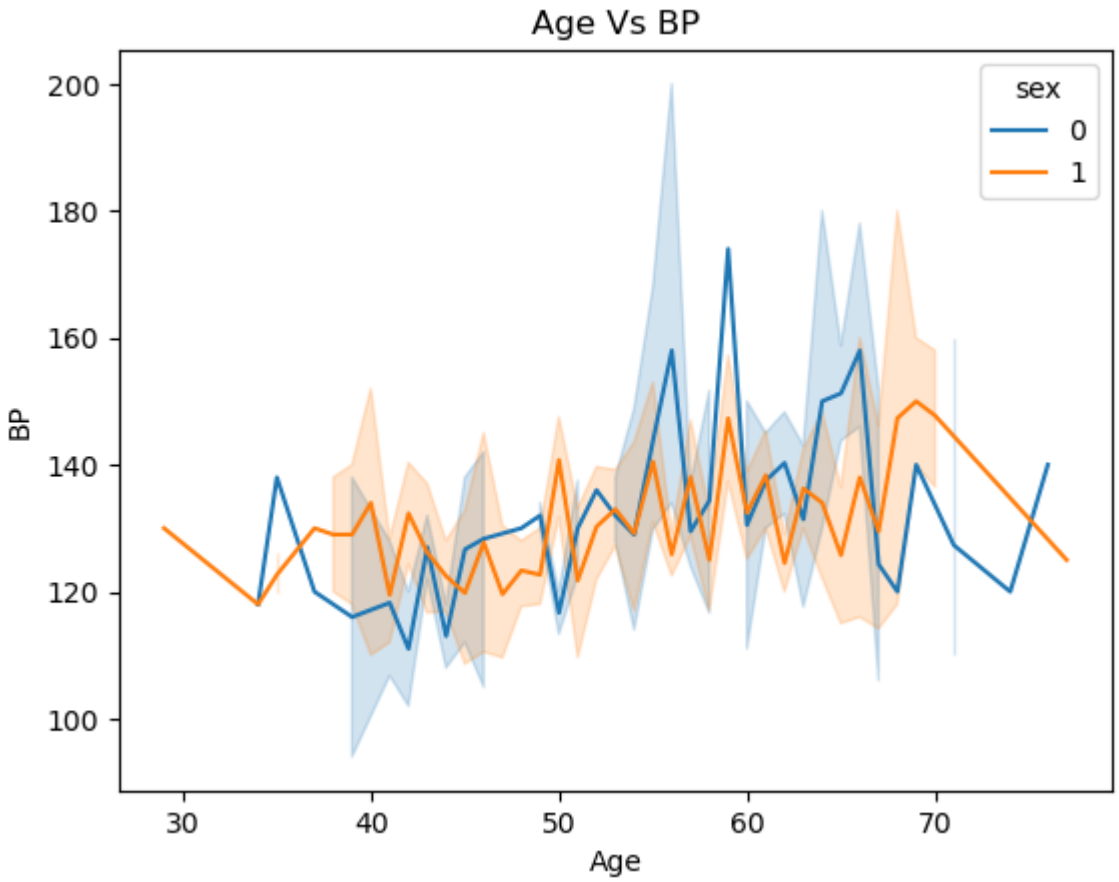
In [92]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='chol_cc',hue='target1')
plt.title('Cholesterol levels Vs Disease')
plt.xlabel('Disease Status')
plt.ylabel('No.Of People')
```



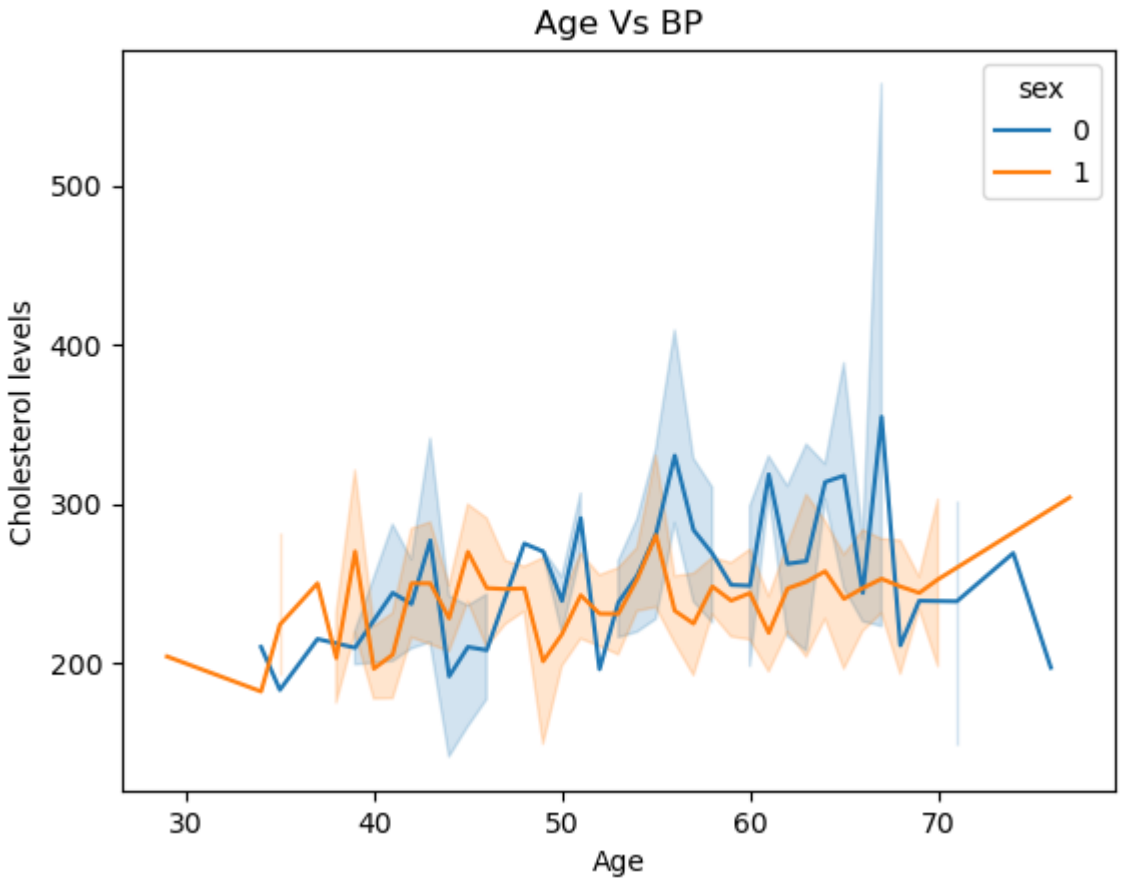
In [93]:
```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='chol_cc',hue='sex')
plt.title('Cholesterol levels Vs Gender')
plt.xlabel('Cholesterol levels')
plt.ylabel('No.Of People')
```



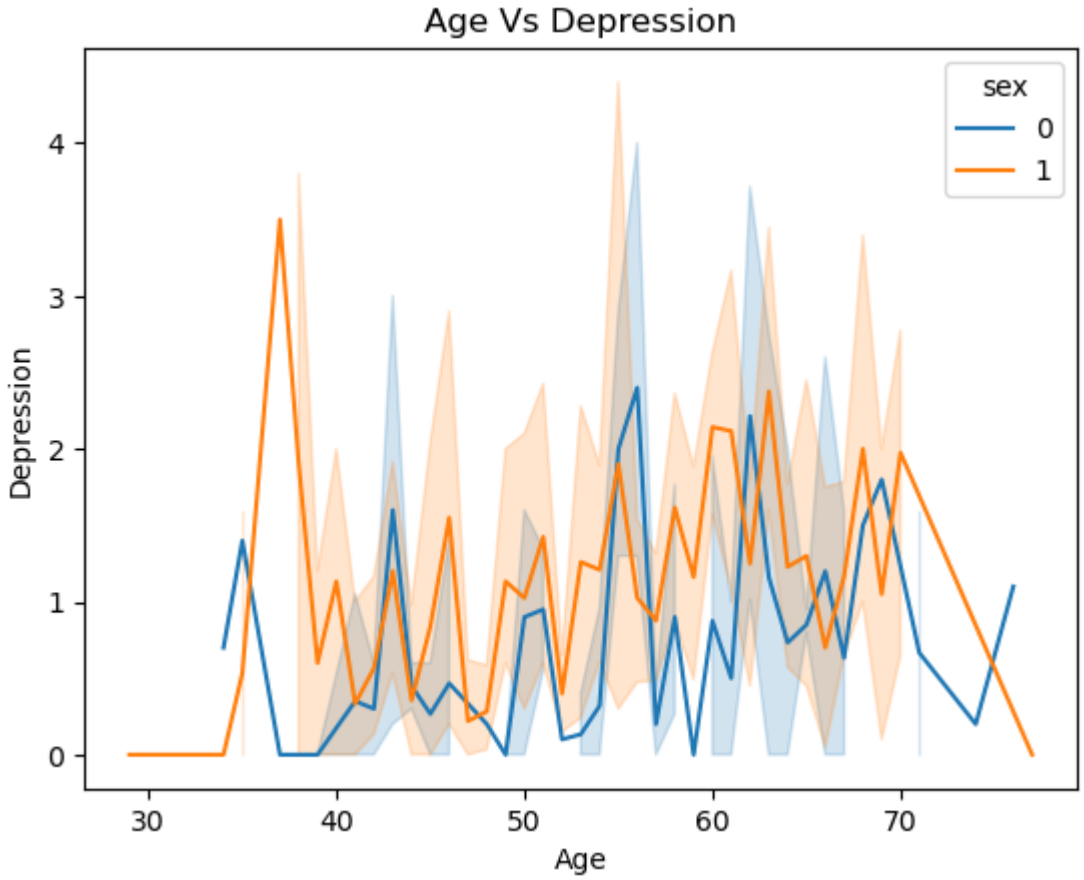
In [94]:
```
sns.lineplot(data=df,x='age',y='trestbps',hue='sex')
plt.title('Age Vs BP')
plt.xlabel('Age')
plt.ylabel('BP')
```



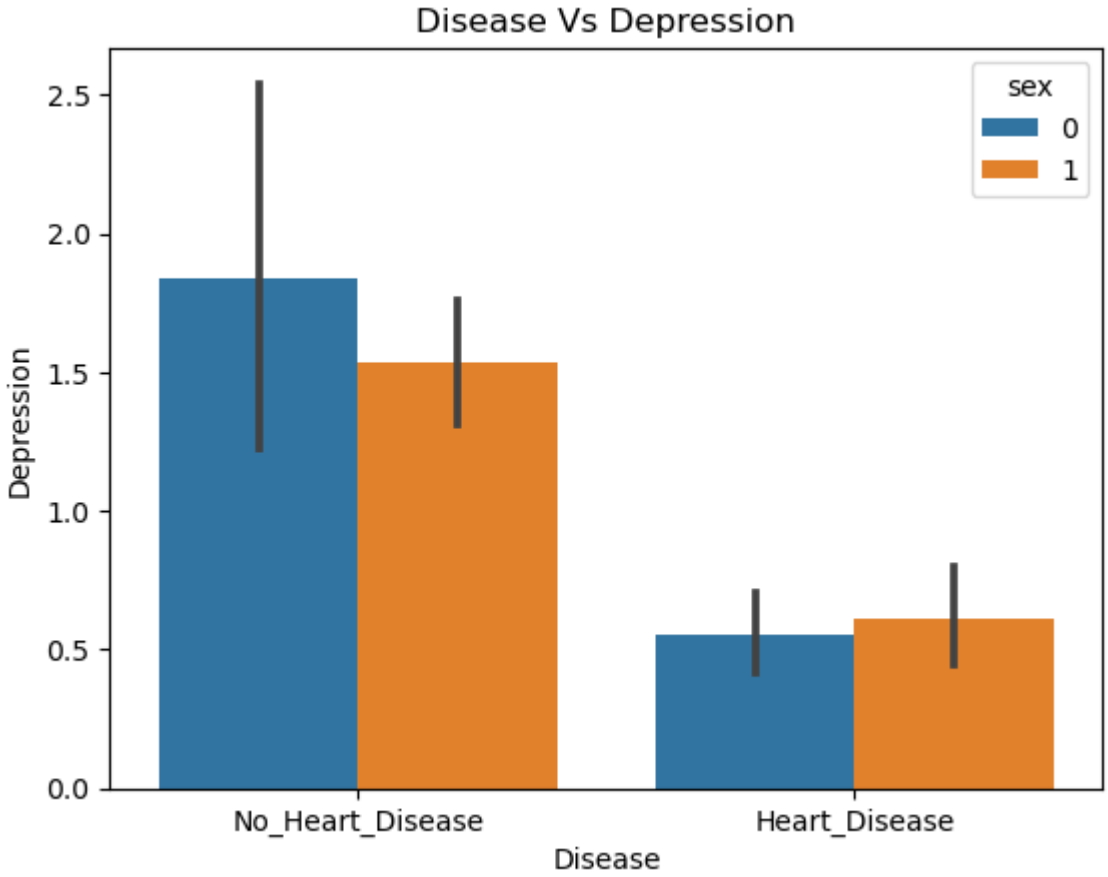
In [95]:
```
sns.lineplot(data=df,x='age',y='chol',hue='sex')
plt.title('Age Vs BP')
plt.xlabel('Age')
plt.ylabel('Cholesterol levels')
```



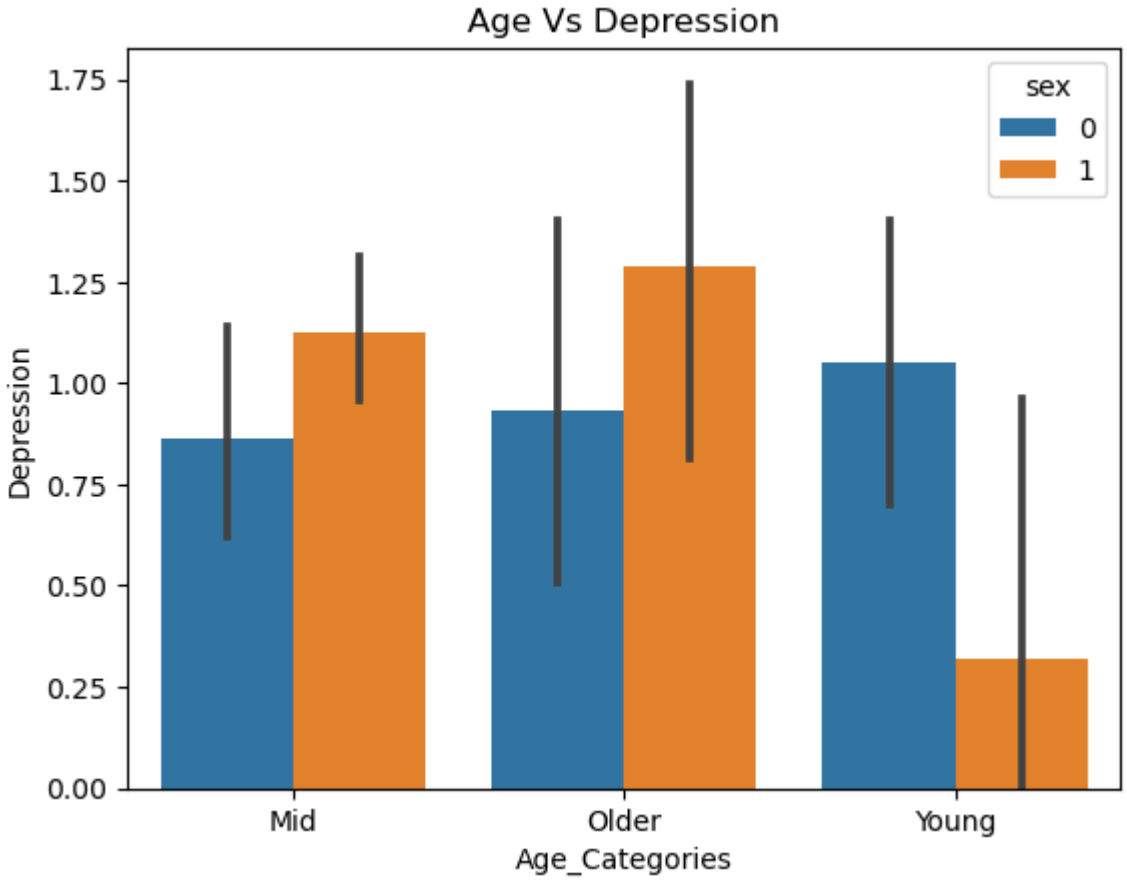
In [96]:
```
sns.lineplot(data=df,x='age',y='oldpeak',hue='sex')
plt.title('Age Vs Depression')
plt.xlabel('Age')
plt.ylabel('Depression')
```



In [97]:
```
sns.barplot(data=df,x='target1',y='oldpeak',hue='sex')
plt.title('Disease Vs Depression')
plt.xlabel('Disease')
plt.ylabel('Depression')
```



In [98]:
```
sns.barplot(data=df,x='age_cat',y='oldpeak',hue='sex')
plt.title('Age Vs Depression')
plt.xlabel('Age_Categories')
plt.ylabel('Depression')
```

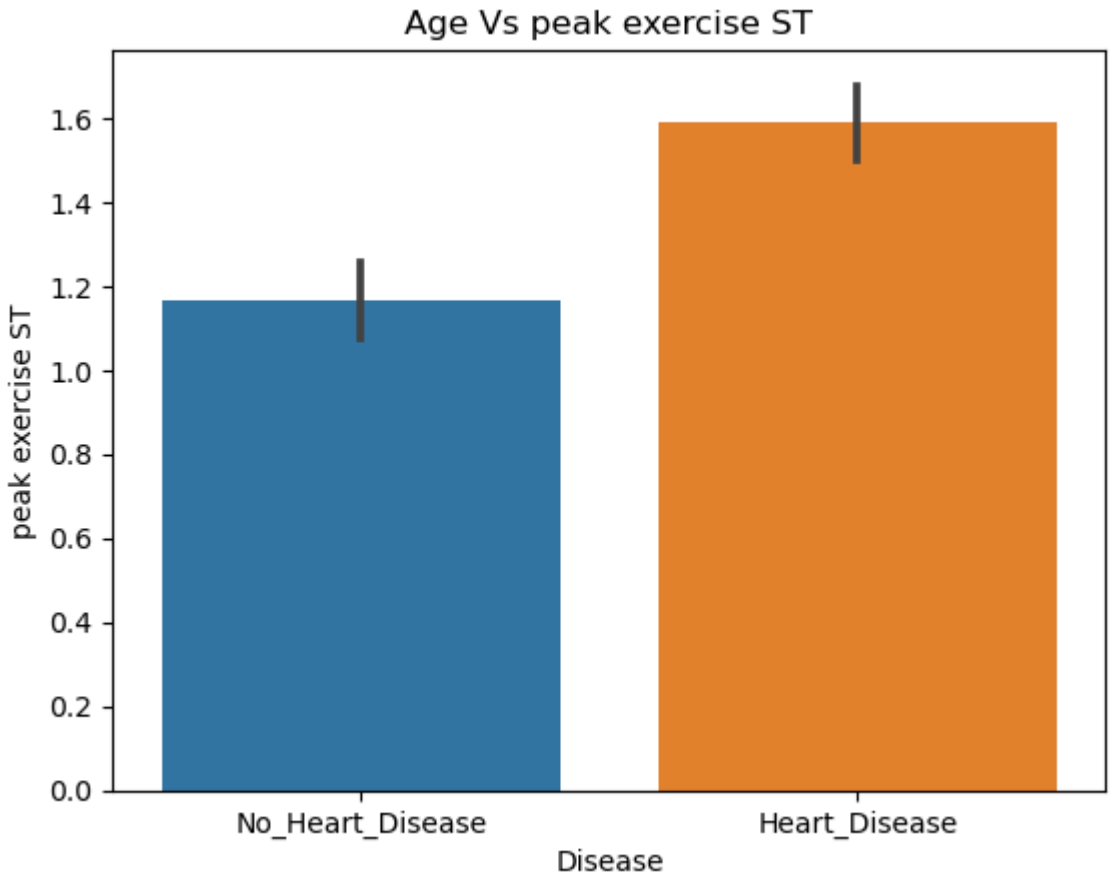


In [99]:
```
df.head()
```

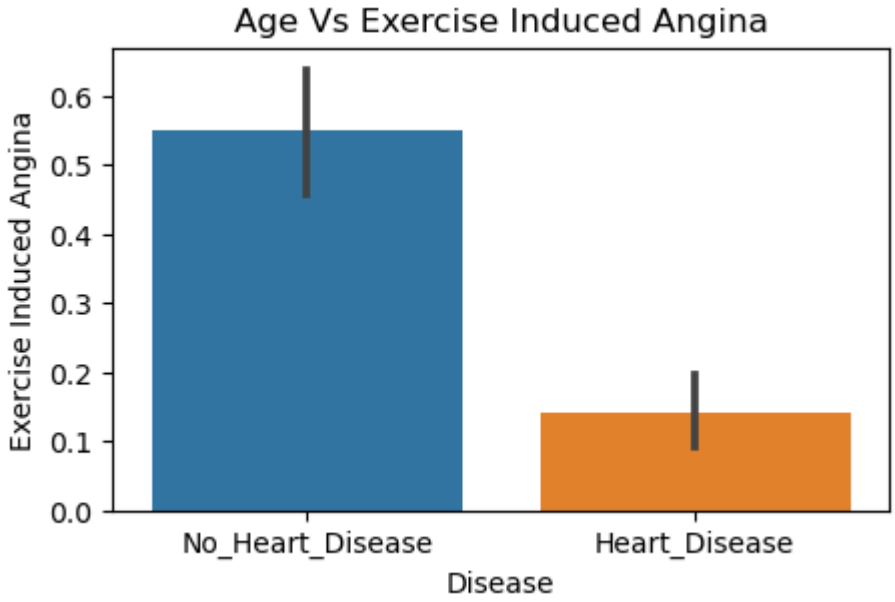
Out[99]:

	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 [100]:
```
sns.barplot(data=df,x='target1',y='slope')
plt.title('Age Vs peak exercise ST')
plt.xlabel('Disease')
plt.ylabel('peak exercise ST')
```

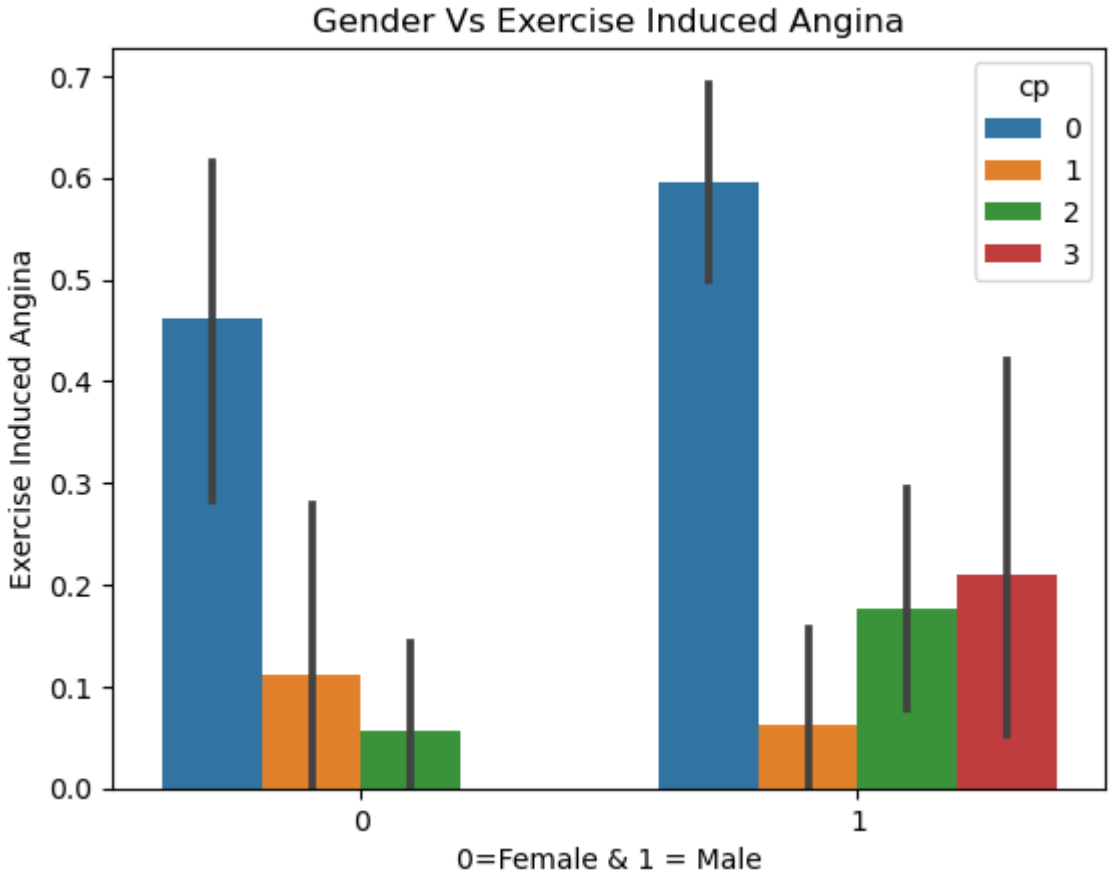


In [116]:
```
plt.figure(figsize=(5,3))
sns.barplot(data=df,x='target1',y='exang') #exercise induced angina
plt.title('Age Vs Exercise Induced Angina')
plt.xlabel('Disease')
plt.ylabel('Exercise Induced Angina')
plt.show()
```



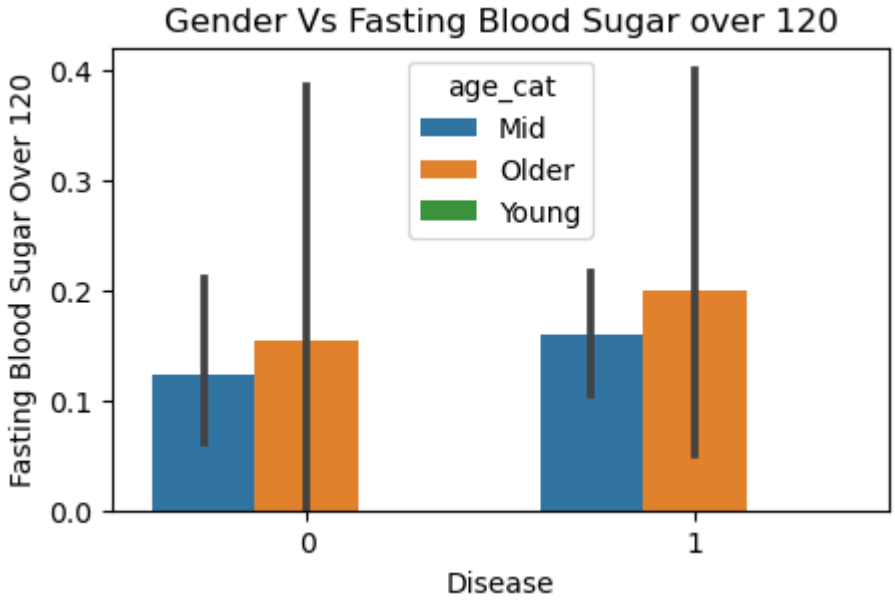
In [102]:

In [103]:
```
sns.barplot(data=df,x='sex',y='exang',hue='cp')
plt.title('Gender Vs Exercise Induced Angina')
plt.xlabel('0=Female & 1 = Male')
plt.ylabel('Exercise Induced Angina')
plt.show()
```



In [104]:
```
# males having high Excerise Angina
```

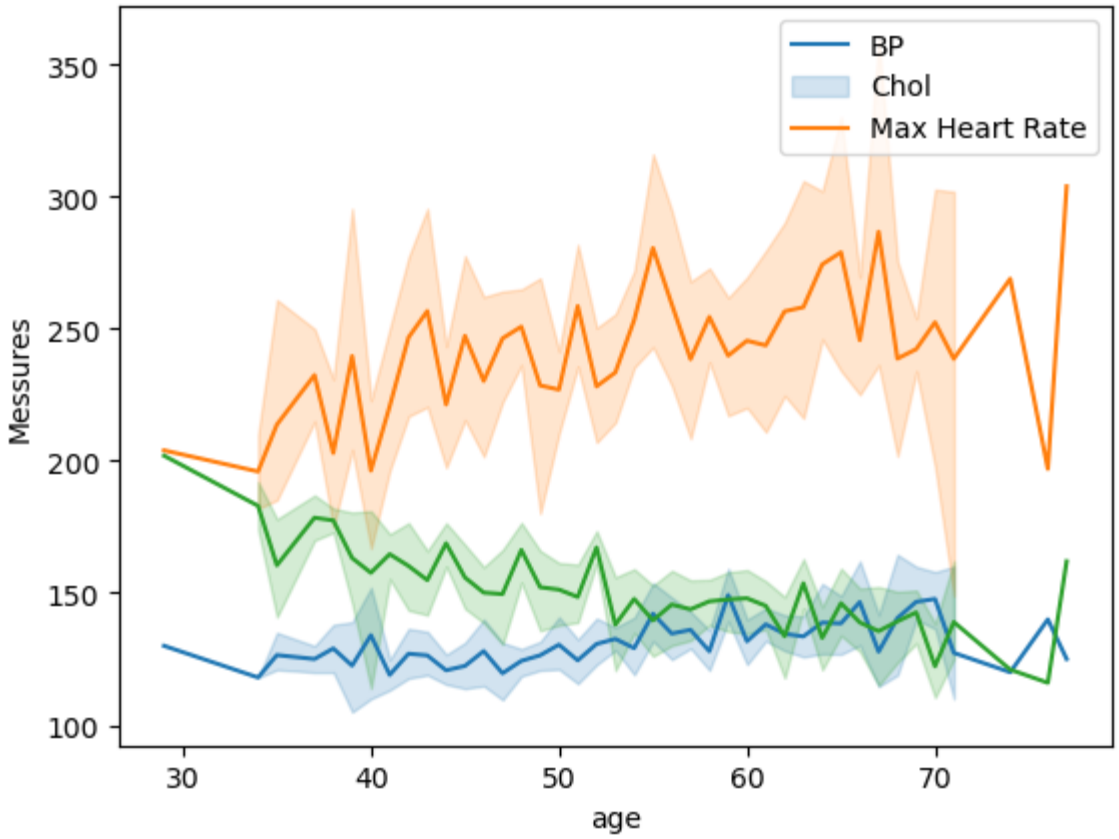
In [117]:
```
plt.figure(figsize=(5,3))
sns.barplot(data=df,x='sex',y='fbs',hue='age_cat')
plt.title('Gender Vs Fasting Blood Sugar over 120')
plt.xlabel('Disease')
plt.ylabel('Fasting Blood Sugar Over 120')
```



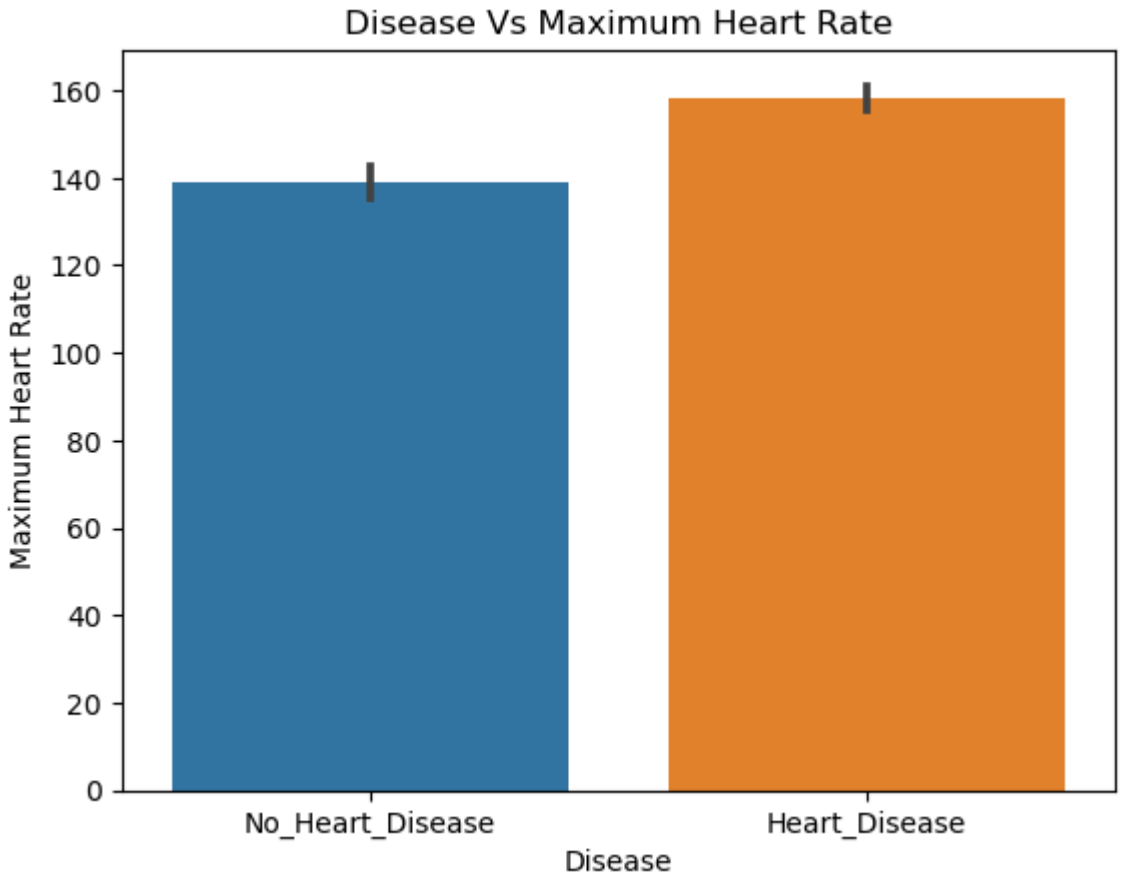
In [106]:

In [107]:
```
sns.lineplot(data=df,x='age',y='trestbps')
sns.lineplot(data=df,x='age',y='chol')
sns.lineplot(data=df,x='age',y='thalach')
plt.ylabel('Messures')
```

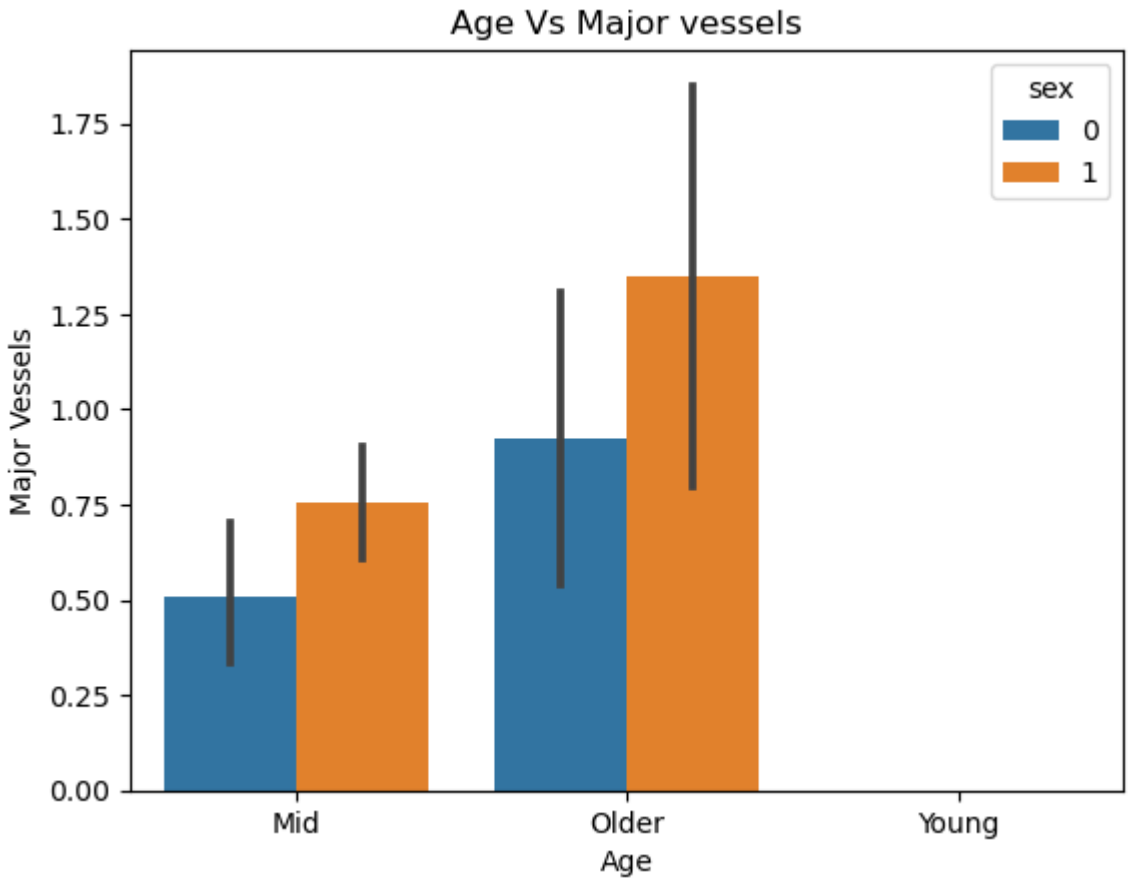
Out[107]: <matplotlib.legend.Legend at 0x23d23511d90>



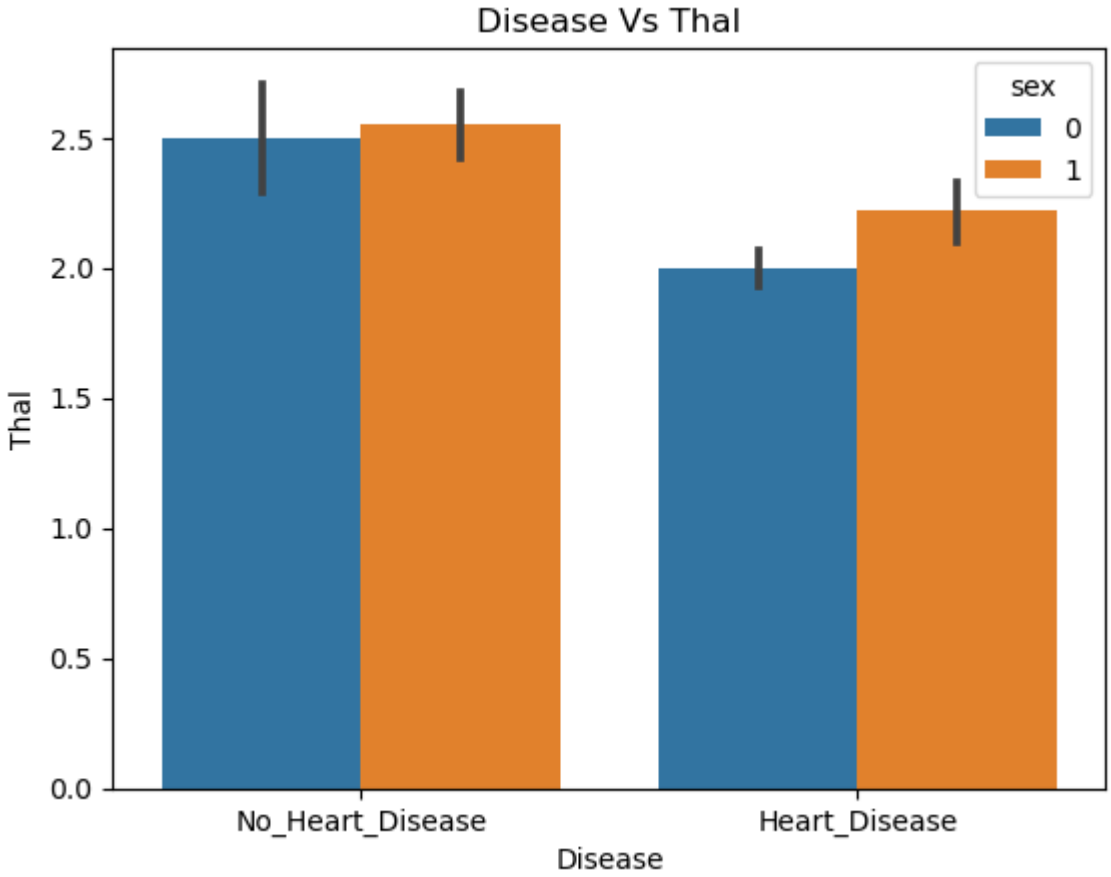
In [108]:
```
sns.barplot(data=df,x='target1',y='thalach')
plt.title('Disease Vs Maximum Heart Rate')
plt.xlabel('Disease')
plt.ylabel('Maximum Heart Rate')
```



In [109]:
```
sns.barplot(data=df,x='age_cat',y='ca',hue='sex')
plt.title('Age Vs Major vessels')
plt.xlabel('Age')
plt.ylabel('Major Vessels')
```



In [110]:
```
sns.barplot(data=df,x='target1',y='thal',hue='sex')
plt.title('Disease Vs Thal')
plt.xlabel('Disease')
plt.ylabel('Thal')
```



In [111]:

In [112]:

Out[112]:

	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 [ ]:

In [ ]: