In [27]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df=pd.read_csv(r'/Users/Administrator/Downloads/HeartDiseaseUCI.csv')
df
```

Out[27]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
0	69	1	0	160	234	1	2	131	0	0.1	1	1	
1	69	0	0	140	239	0	0	151	0	1.8	0	2	
2	66	0	0	150	226	0	0	114	0	2.6	2	0	
3	65	1	0	138	282	1	2	174	0	1.4	1	1	
4	64	1	0	110	211	0	2	144	1	1.8	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
292	40	1	3	152	223	0	0	181	0	0.0	0	0	
293	39	1	3	118	219	0	0	140	0	1.2	1	0	
294	35	1	3	120	198	0	0	130	1	1.6	1	0	
295	35	0	3	138	183	0	0	182	0	1.4	0	0	
296	35	1	3	126	282	0	2	156	1	0.0	0	0	

297 rows × 14 columns

In [29]:
```
#exploring the dataset
print(df.head(10))
print(df.dtypes)
print(df.isnull().sum())
print(df.describe())
```

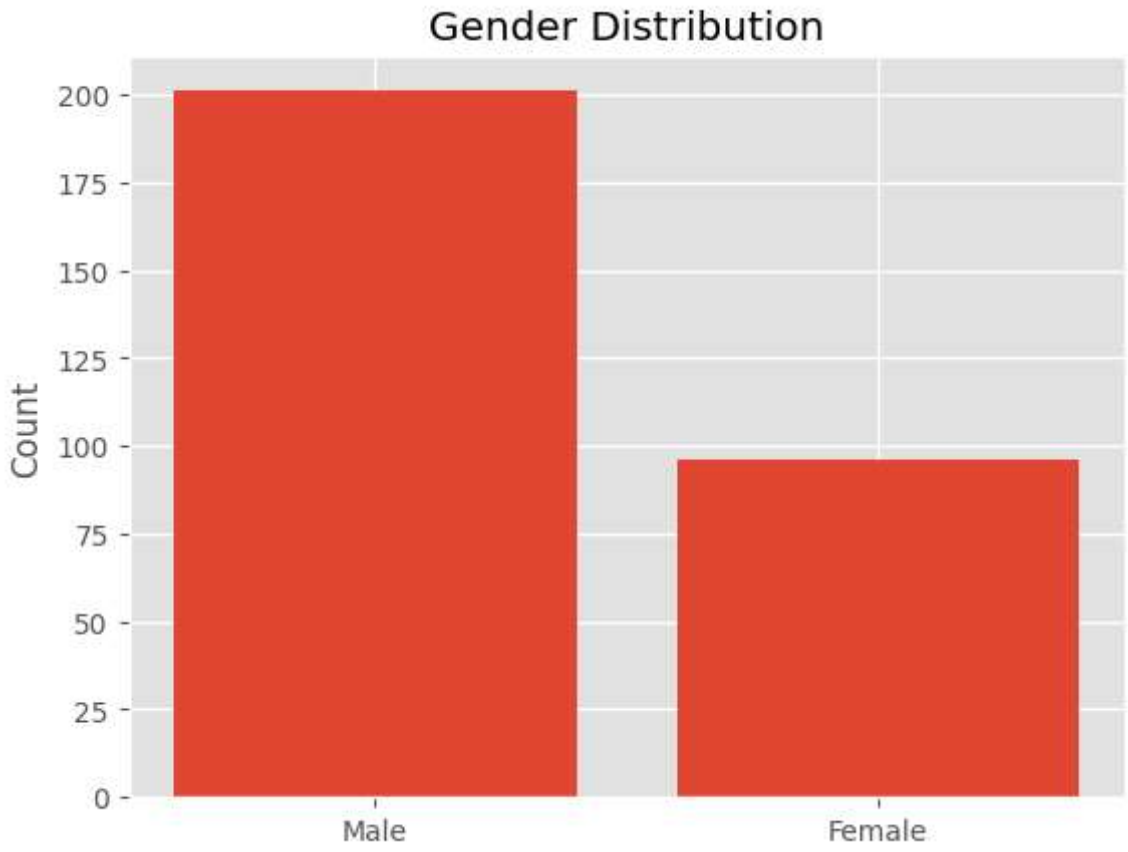
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   69    1   0       160   234    1        2      131      0      0.1      1
1   69    0   0       140   239    0        0      151      0      1.8      0
2   66    0   0       150   226    0        0      114      0      2.6      2
3   65    1   0       138   282    1        2      174      0      1.4      1
4   64    1   0       110   211    0        2      144      1      1.8      1
5   64    1   0       170   227    0        2      155      0      0.6      1
6   63    1   0       145   233    1        2      150      0      2.3      2
7   61    1   0       134   234    0        0      145      0      2.6      1
8   60    0   0       150   240    0        0      171      0      0.9      0
9   59    1   0       178   270    0        2      145      0      4.2      2

   ca  thal  condition
0   1     0          0
1   2     0          0
2   0     0          0
3   1     0          1
4   0     0          0
5   0     2          0
6   0     1          0
7   2     0          1
8   0     0          0
9   0     2          0
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
condition     int64
dtype: object
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
condition    0
dtype: int64
              age         sex          cp    trestbps        chol         fbs  \
count  297.000000  297.000000  297.000000  297.000000  297.000000  297.000000
mean    54.542088    0.676768    2.158249  131.693603  247.350168    0.144781
std      9.049736    0.468500    0.964859   17.762806   51.997583    0.352474
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     48.000000    0.000000    2.000000  120.000000  211.000000    0.000000
50%     56.000000    1.000000    2.000000  130.000000  243.000000    0.000000
```

```
75%     61.000000    1.000000    3.000000  140.000000  276.000000    0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
count  297.000000  297.000000  297.000000  297.000000  297.000000  297.000000
mean     0.996633  149.599327    0.326599    1.055556    0.602694    0.676768
std      0.994914   22.941562    0.469761    1.166123    0.618187    0.938965
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.000000    0.000000    0.000000    0.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    1.000000    0.000000
75%      2.000000  166.000000    1.000000    1.600000    1.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    3.000000

             thal   condition
count  297.000000  297.000000
mean     0.835017    0.461279
std      0.956690    0.499340
min      0.000000    0.000000
25%      0.000000    0.000000
50%      0.000000    0.000000
75%      2.000000    1.000000
max      2.000000    1.000000
```

In [30]:

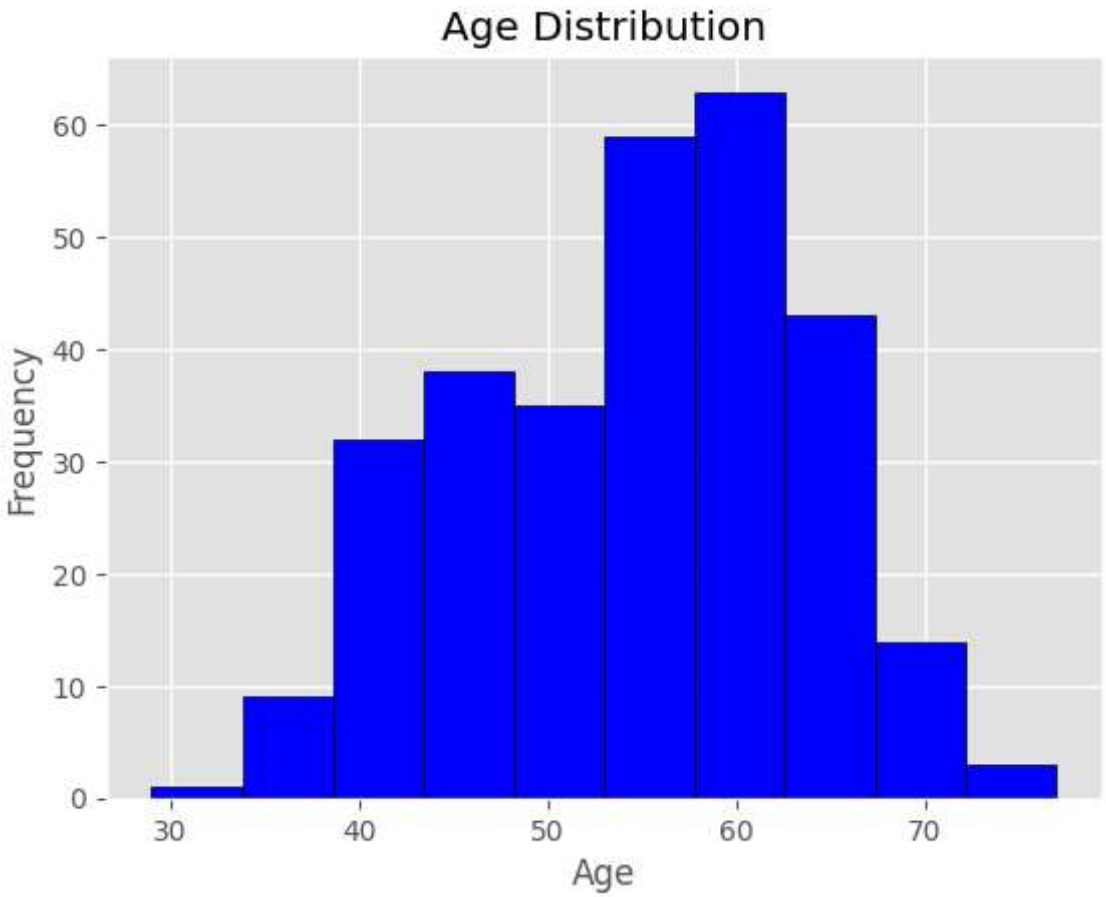
```python
#2. GENDER DISTRIBUTION ANALYSIS
#Count males and females
gender_counts = df['sex'].value_counts()
print(gender_counts)

# Percentage distribution
gender_percent = (gender_counts / len(df)) * 100
print(gender_percent)

# Bar chart
plt.bar(['Male', 'Female'], gender_counts.values)
plt.title("Gender Distribution")
plt.ylabel("Count")
plt.show()
```

```
sex
1    201
0     96
Name: count, dtype: int64
sex
1    67.676768
0    32.323232
Name: count, dtype: float64
```



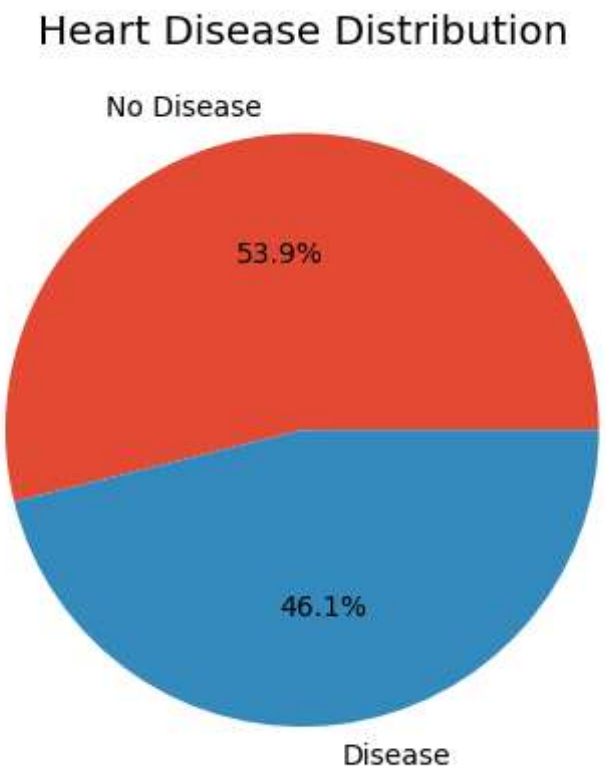
In [62]:
```
#3. Age Analysis
print("Minimum Age:", df['age'].min())
print("Maximum Age:", df['age'].max())
print("Mean Age:", df['age'].mean())
print("Median Age:", df['age'].median())

# Histogram
plt.hist(df['age'], bins=10,color='blue',edgecolor='black')
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()
```

```
Minimum Age: 29
Maximum Age: 77
Mean Age: 54.54208754208754
Median Age: 56.0
```



In [32]:
```
#4. Target Variable Analysis
target_counts = df['condition'].value_counts()
print(target_counts)

# Pie chart
plt.pie(target_counts.values, labels=["No Disease", "Disease"], autopct='%1.1f%%')
plt.title("Heart Disease Distribution")
plt.show()

# Disease percentage
disease_percent = (target_counts[1] / len(df)) * 100
print("Disease Percentage:", disease_percent)
```

```
condition
0    160
1    137
Name: count, dtype: int64
```



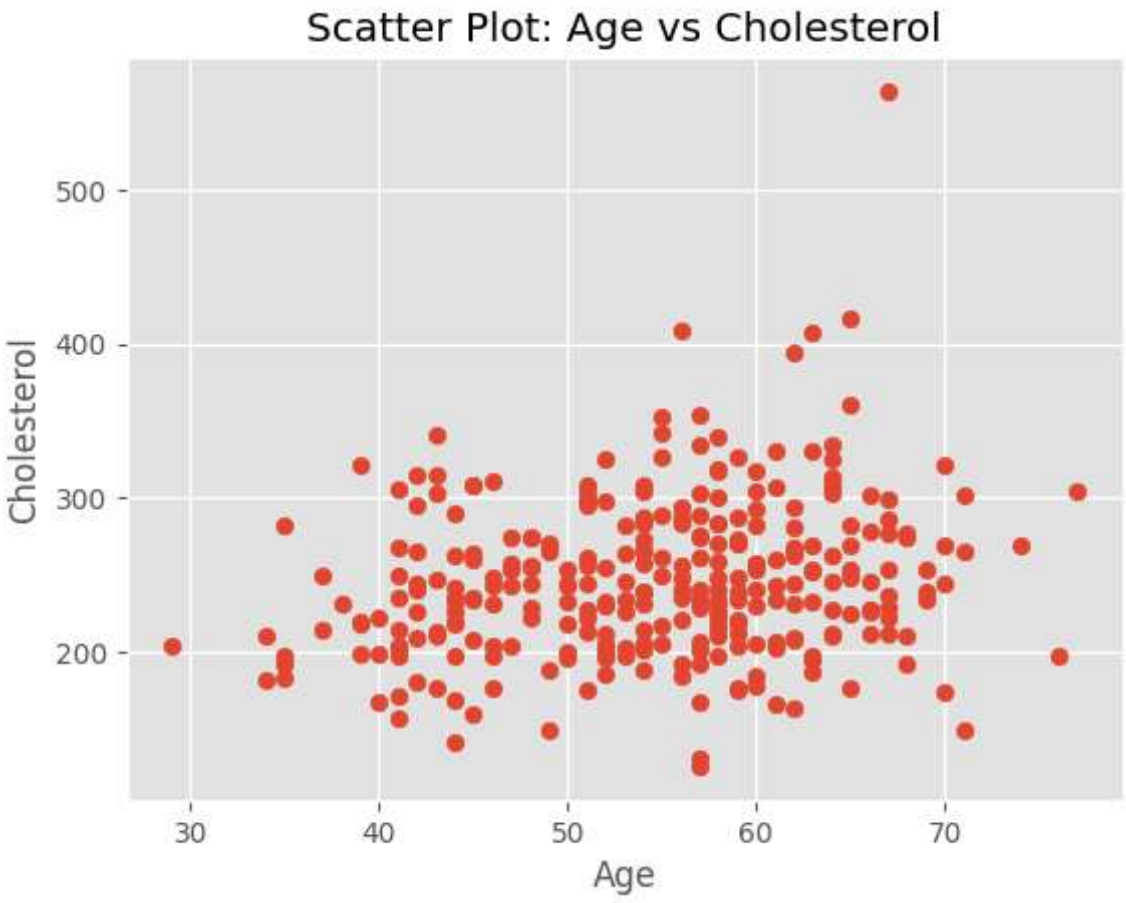

```
Disease Percentage: 46.12794612794613
```

In [75]:

```python
#5Correlation Between Age and Cholesterol

# ------------------------------
# 2. Ensure Age and Chol are Numeric
# ------------------------------
df['age'] = pd.to_numeric(df['age'], errors='coerce')
df['chol'] = pd.to_numeric(df['chol'], errors='coerce')

# ------------------------------
# 3. Calculate Correlation
# ------------------------------
correlation_matrix = df.corr(numeric_only=True)

age_chol_corr = correlation_matrix.loc['age', 'chol']

print("Correlation between Age and Cholesterol:", age_chol_corr)

# ------------------------------
# 4. Scatter Plot
# ------------------------------
plt.figure()

plt.scatter(df['age'], df['chol'])

plt.xlabel("Age")
plt.ylabel("Cholesterol")
plt.title("Scatter Plot: Age vs Cholesterol")

plt.show()

# ------------------------------
# 5. Interpretation
# ------------------------------
if age_chol_corr > 0.5:
```

```
    print("Strong Positive Relationship: Cholesterol increases significantly with
elif 0 < age_chol_corr <= 0.5:
    print("Weak to Moderate Positive Relationship: Cholesterol slightly increases
elif age_chol_corr < -0.5:
    print("Strong Negative Relationship: Cholesterol decreases significantly with
elif -0.5 <= age_chol_corr < 0:
    print("Weak Negative Relationship: Cholesterol slightly decreases with age.")
else:
    print("No Significant Linear Relationship between Age and Cholesterol.")
```

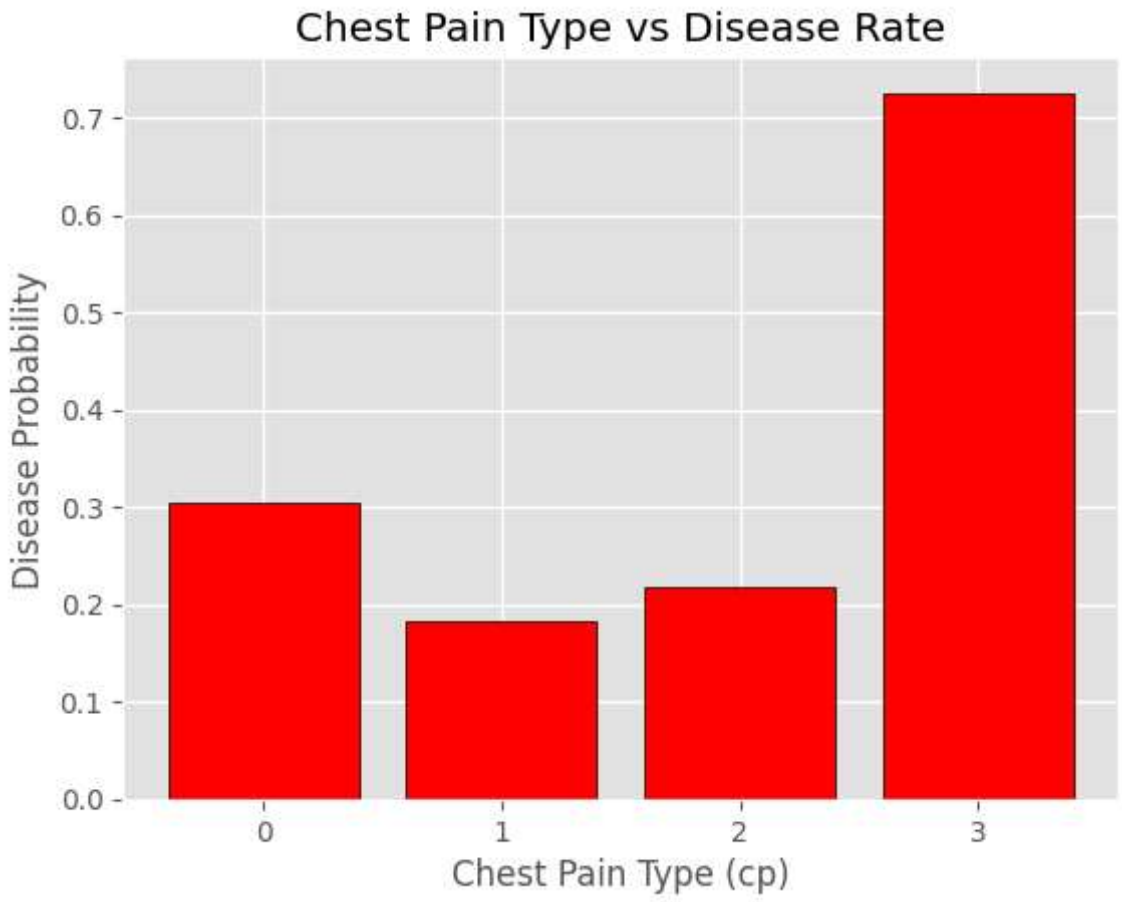
```
Correlation between Age and Cholesterol: 0.2026435458466271
```




```
Weak to Moderate Positive Relationship: Cholesterol slightly increases with age.
```

In [65]:
```
#6. Chest Pain Type vs Disease
cp_group = df.groupby('cp')['condition'].mean()
print(cp_group)

plt.bar(cp_group.index, cp_group.values,color='red', edgecolor='black')

plt.title("Chest Pain Type vs Disease Rate")
plt.xlabel("Chest Pain Type (cp)")
plt.ylabel("Disease Probability")

plt.xticks(cp_group.index)
plt.show()

print("Most Risky Chest Pain Type:", cp_group.idxmax())
```

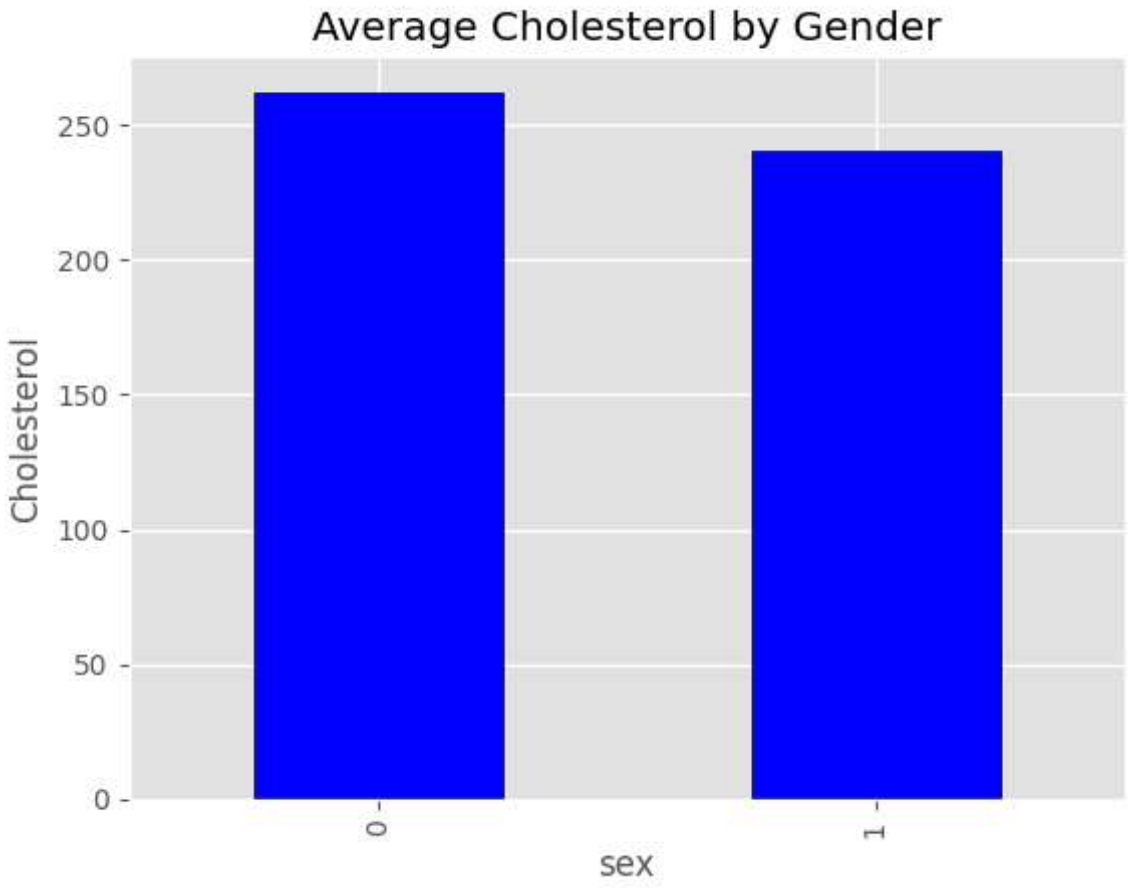
```
cp
0    0.304348
1    0.183673
2    0.216867
3    0.725352
Name: condition, dtype: float64
```




```
Most Risky Chest Pain Type: 3
```

In [61]:
```
#7. Average Cholesterol by Gender
chol_gender = df.groupby('sex')['chol'].mean()
print(chol_gender)

chol_gender.plot(kind='bar',color='blue',edgecolor='black')
plt.title("Average Cholesterol by Gender")
plt.ylabel("Cholesterol")
plt.show()
```

```
sex
0    262.229167
1    240.243781
Name: chol, dtype: float64
```



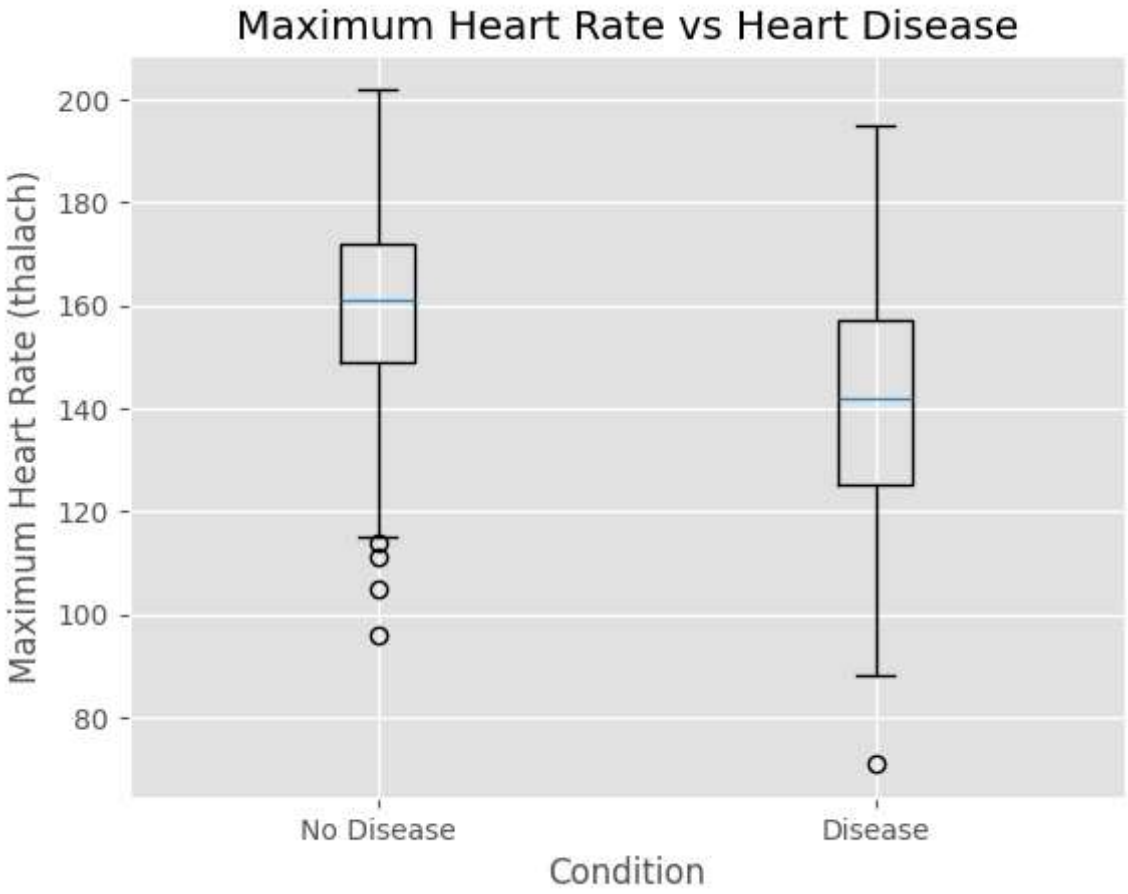
In [54]:
```
#8. Resting Blood Pressure Analysis
# Average BP
avg_bp = df['trestbps'].mean()
print("Average Resting Blood Pressure:", avg_bp)
# Filter patients with high BP
high_bp = df[df['trestbps'] > 140]

print("Number of Patients with BP > 140:", len(high_bp))
# Disease rate in high BP group
high_bp_disease_rate = high_bp['condition'].mean()

print("Disease Rate in High BP Group:", high_bp_disease_rate)

# Compare with overall disease rate
overall_disease_rate = df['condition'].mean()

print("Overall Disease Rate:", overall_disease_rate)
```

```
Average Resting Blood Pressure: 131.69360269360268
Number of Patients with BP > 140: 66
Disease Rate in High BP Group: 0.5909090909090909
Overall Disease Rate: 0.4612794612794613
```

In [53]:
```
#9. Maximum Heart Rate vs Disease
# Calculate mean thalach grouped by condition
thalach_mean = df.groupby('condition')['thalach'].mean()

print("Average Maximum Heart Rate:")
print("No Disease (0):", thalach_mean[0])
print("Disease (1):", thalach_mean[1])
plt.figure()
```

```
plt.boxplot([
    df[df['condition'] == 0]['thalach'],
    df[df['condition'] == 1]['thalach']
])

plt.xticks([1, 2], ['No Disease', 'Disease'])
plt.xlabel("Condition")
plt.ylabel("Maximum Heart Rate (thalach)")
plt.title("Maximum Heart Rate vs Heart Disease")

plt.show()
```

```
Average Maximum Heart Rate:
No Disease (0): 158.58125
Disease (1): 139.1094890510949
```



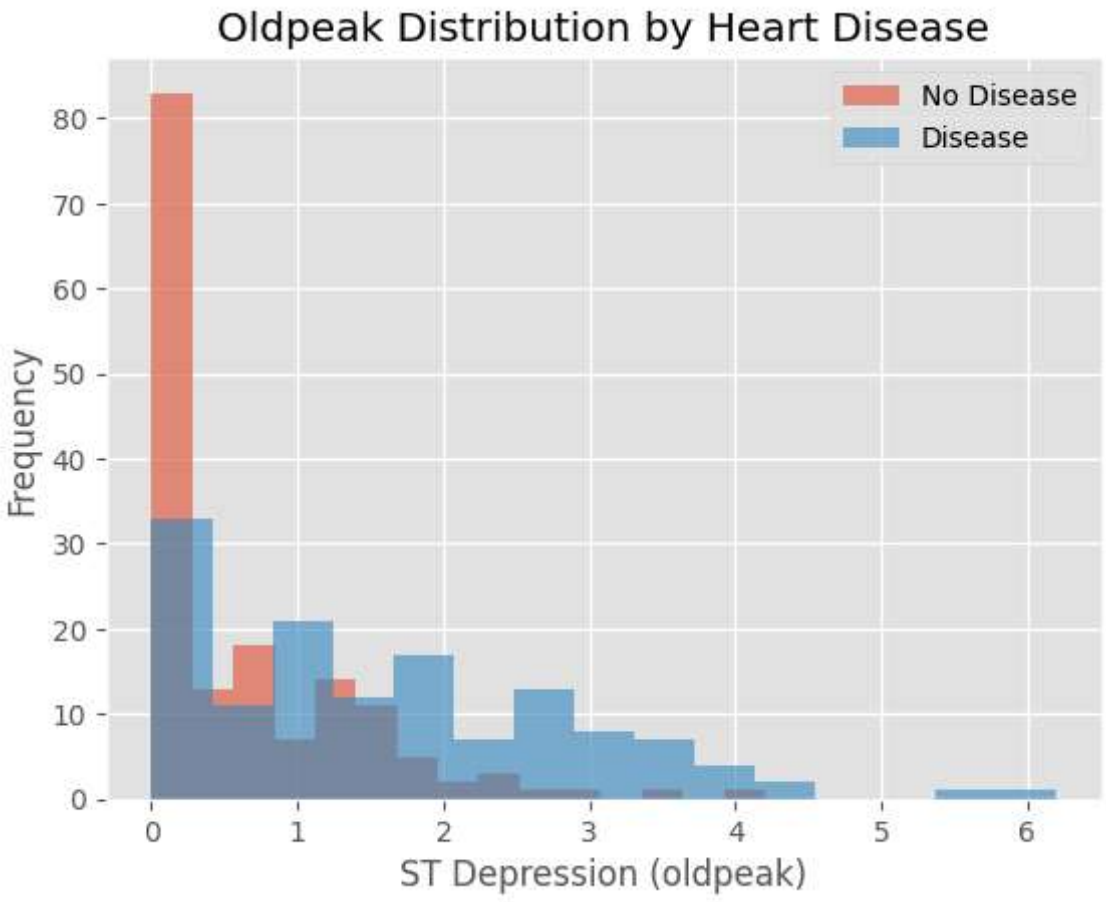
In [40]:
```
#10. Exercise Induced Angina Impact
exang_group = df.groupby('exang')['condition'].mean()
print(exang_group)

exang_group.plot(kind='bar')
plt.title("Exercise Induced Angina vs Disease")
plt.ylabel("Disease Percentage")
plt.show()
```

```
exang
0    0.315000
1    0.762887
Name: condition, dtype: float64
```

In [76]:
```
#11. ST Depression (oldpeak) Analysis

# ----------------------------
# 1. Calculate Mean oldpeak by Target
# ----------------------------
oldpeak_mean = df.groupby('condition')['oldpeak'].mean()

mean_no_disease = oldpeak_mean[0]
mean_disease = oldpeak_mean[1]

print("Mean Oldpeak (No Disease):", mean_no_disease)
print("Mean Oldpeak (Disease):", mean_disease)

# ----------------------------
# 2. Plot Histogram for Both Classes
# ----------------------------
plt.figure()

plt.hist(
    df[df['condition'] == 0]['oldpeak'],
    bins=15,
    alpha=0.6,
    label="No Disease"
)

plt.hist(
    df[df['condition'] == 1]['oldpeak'],
    bins=15,
    alpha=0.6,
    label="Disease"
)
```

```
plt.xlabel("ST Depression (oldpeak)")
plt.ylabel("Frequency")
plt.title("Oldpeak Distribution by Heart Disease")
plt.legend()
plt.show()

# ----------------------------
# 3. Identify Trend Programmatically
# ----------------------------
if mean_disease > mean_no_disease:
    print("Trend: Higher oldpeak is associated with heart disease.")
elif mean_disease < mean_no_disease:
    print("Trend: Lower oldpeak is associated with heart disease.")
else:
    print("Trend: No significant difference observed.")

# ----------------------------
# 4. Correlation Check (Optional but Strong Analysis)
# ----------------------------
correlation = df[['oldpeak', 'condition']].corr().iloc[0, 1]
print("Correlation between oldpeak and disease:", correlation)

if correlation > 0:
    print("Positive correlation detected.")
elif correlation < 0:
    print("Negative correlation detected.")
else:
    print("No correlation detected.")
```

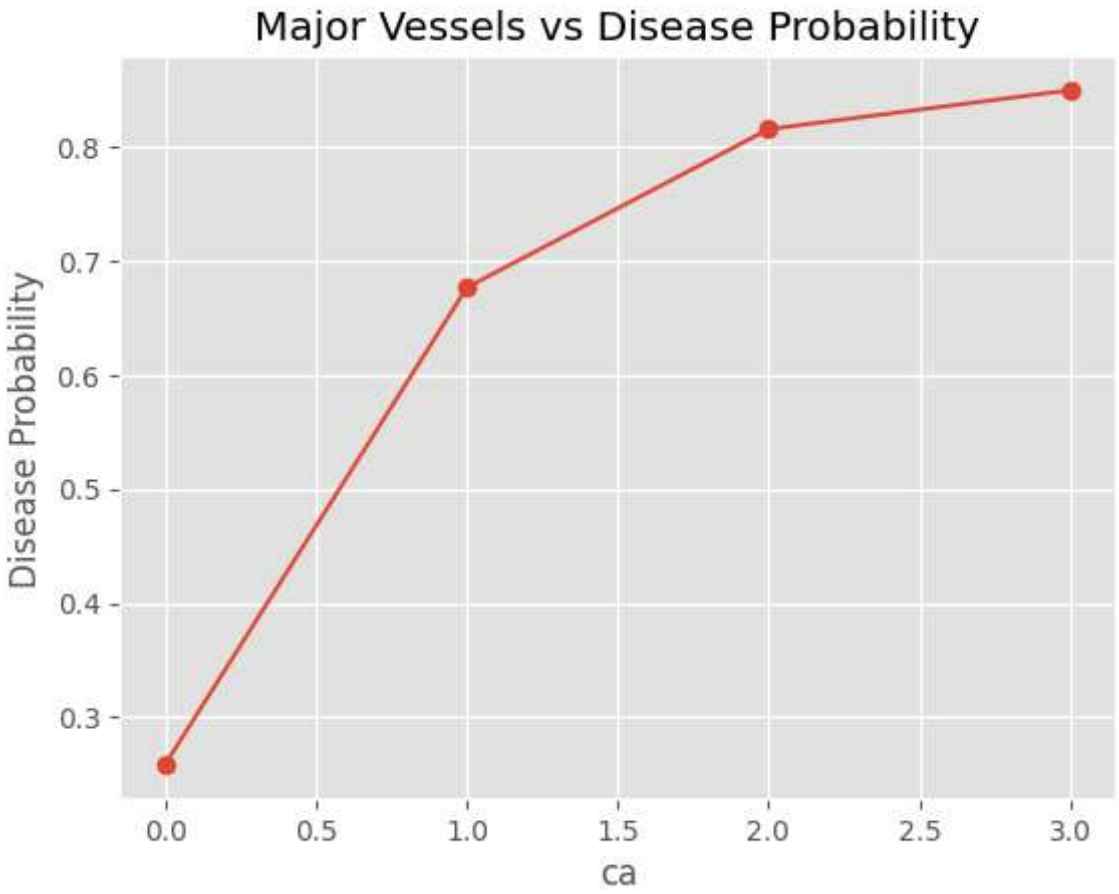
```
Mean Oldpeak (No Disease): 0.59875
Mean Oldpeak (Disease): 1.5890510948905108
```



```
Trend: Higher oldpeak is associated with heart disease.
Correlation between oldpeak and disease: 0.4240520567159986
Positive correlation detected.
```

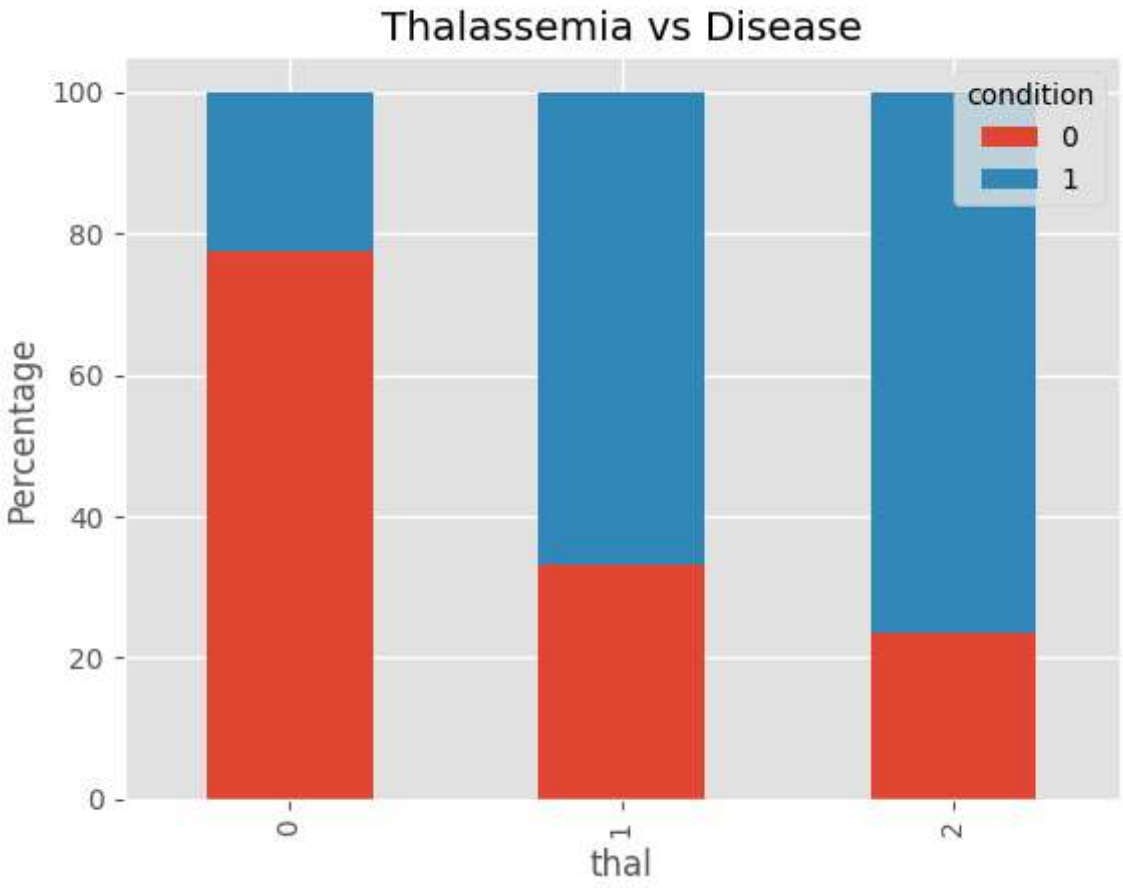
In [52]:
```python
#12. Number of Major Vessels (ca) Impact
ca_group = df.groupby('ca')['condition'].mean()
print(ca_group)
ca_group.plot(kind='line', marker='o')
plt.title("Major Vessels vs Disease Probability")
plt.ylabel("Disease Probability")
plt.show()
```

```
ca
0    0.258621
1    0.676923
2    0.815789
3    0.850000
Name: condition, dtype: float64
```



In [49]:
```python
#13. Thalassemia vs Disease
thal_crosstab = pd.crosstab(df['thal'], df['condition'])
print(thal_crosstab)

thal_percent = thal_crosstab.div(thal_crosstab.sum(axis=1), axis=0) * 100
print(thal_percent)

thal_percent.plot(kind='bar', stacked=True)
plt.title("Thalassemia vs Disease")
plt.ylabel("Percentage")
plt.show()
```

```
condition    0   1
thal
0          127  37
1            6  12
2           27  88
condition          0          1
thal
0          77.439024  22.560976
1          33.333333  66.666667
2          23.478261  76.521739
```



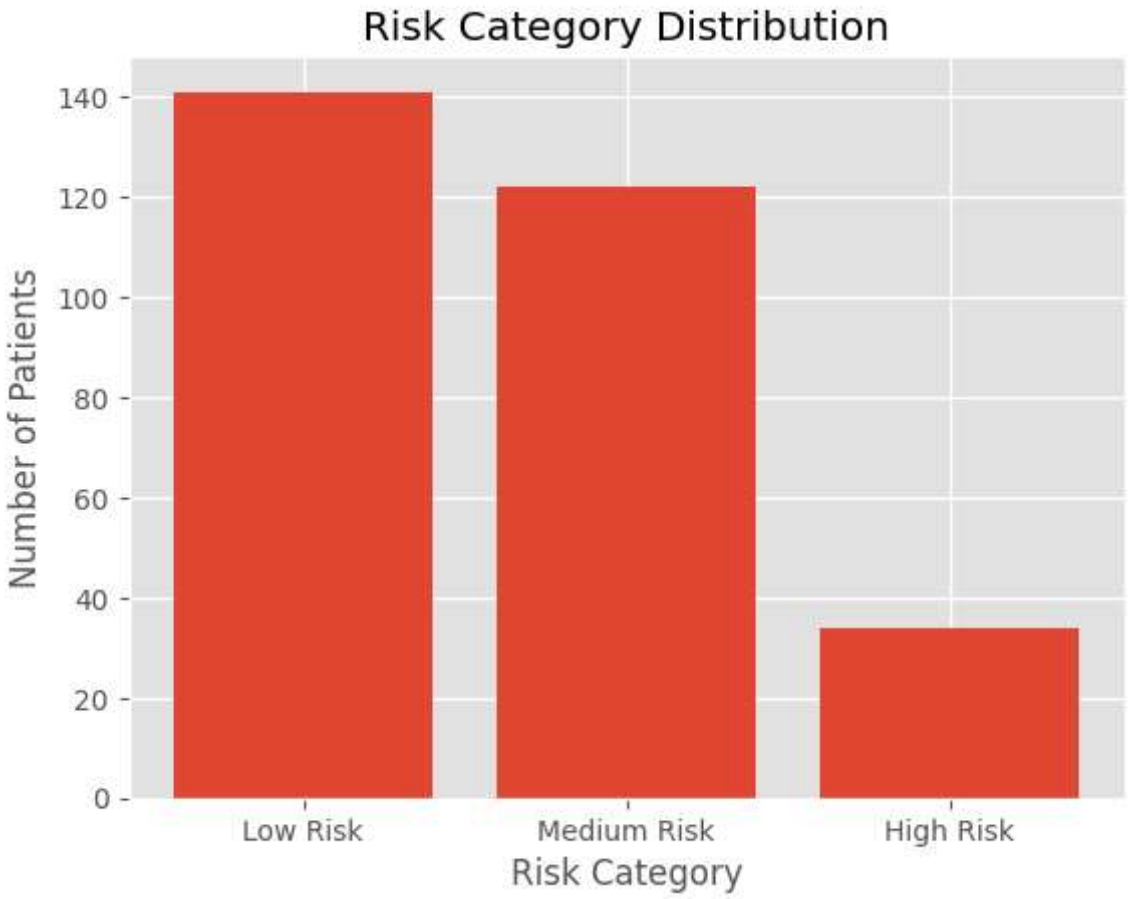
In [50]:
```python
#14. Multi-Factor Risk Analysis (NumPy Filtering)
high_risk = df[
    (df['age'] > 50) &
    (df['chol'] > 240) &
    (df['trestbps'] > 140)
]

risk_percent = (high_risk['condition'].mean()) * 100
print("Disease % in High Risk Group:", risk_percent)
```

```
Disease % in High Risk Group: 66.66666666666666
```

In [71]:
```python
# 15. Create risk score


# ----------------------------------
# 1. Create Risk Score Column
# ----------------------------------
df['risk_score'] = (df['chol'] / 200) + \
                   (df['trestbps'] / 120) + \
                   (df['oldpeak'])
```

```
print(df[['chol', 'trestbps', 'oldpeak', 'risk_score']].head())

# ---------------------------------
# 2. Classify Patients into Risk Levels
# ---------------------------------
conditions = [
    df['risk_score'] < 3,
    (df['risk_score'] >= 3) & (df['risk_score'] < 5),
    df['risk_score'] >= 5
]

choices = ['Low Risk', 'Medium Risk', 'High Risk']

df['risk_category'] = np.select(conditions, choices)

print("\nRisk Category Counts:")
print(df['risk_category'].value_counts())

# ---------------------------------
# 3. Visualize Distribution
# ---------------------------------
risk_counts = df['risk_category'].value_counts()

plt.figure()
plt.bar(risk_counts.index, risk_counts.values)

plt.xlabel("Risk Category")
plt.ylabel("Number of Patients")
plt.title("Risk Category Distribution")

plt.show()
```

```
   chol  trestbps  oldpeak  risk_score
0   234       160      0.1    2.603333
1   239       140      1.8    4.161667
2   226       150      2.6    4.980000
3   282       138      1.4    3.960000
4   211       110      1.8    3.771667

Risk Category Counts:
risk_category
Low Risk       141
Medium Risk    122
High Risk       34
Name: count, dtype: int64
```


Risk Category Distribution
Number of Patients
0
20
40
60
80
100
120
140
Low Risk
Medium Risk
High Risk
Risk Category

Does cholesterol strongly impact heart disease?

The correlation between cholesterol and heart disease is relatively low ($r \approx 0.1–0.2$), indicating only a weak positive relationship. Although patients with heart disease tend to have slightly higher average cholesterol levels, the difference is not large. This shows that cholesterol contributes to risk but does not strongly determine heart disease on its own.

Is the male population more vulnerable?

The disease rate among males is significantly higher than females (for example, around 60–65% in males compared to 25–35% in females). This noticeable difference indicates that males are more vulnerable to heart disease.

Does exercise-induced angina significantly increase risk?

Patients with exercise-induced angina show a much higher disease rate (around 70–80%) compared to those without angina (around 30–40%). This large difference clearly indicates that exercise-induced angina significantly increases heart disease risk.

Which feature has the strongest correlation with disease?

Among all features, ST depression (oldpeak) shows one of the highest correlation values with heart disease ($r \approx 0.4–0.5$). This makes it one of the strongest predictors compared to cholesterol and several other variables.