In [7]:
```
import numpy as np
import pandas as pd

# Set random seed for reproducibility
np.random.seed(42)

# Generate synthetic data
num_rows = 500
age = np.random.randint(29, 80, num_rows)
sex = np.random.choice([0, 1], num_rows)  # 0 = female, 1 = male
cp = np.random.randint(0, 4, num_rows)  # Chest pain type
trestbps = np.random.randint(94, 200, num_rows)  # Resting blood pressure
chol = np.random.randint(126, 564, num_rows)  # Serum cholesterol
fbs = np.random.choice([0, 1], num_rows)  # Fasting blood sugar > 120 mg/dl
restecg = np.random.randint(0, 3, num_rows)  # Resting ECG results
thalach = np.random.randint(71, 202, num_rows)  # Max heart rate achieved
exang = np.random.choice([0, 1], num_rows)  # Exercise-induced angina
oldpeak = np.round(np.random.uniform(0.0, 6.2, num_rows), 1)  # ST depression
slope = np.random.randint(0, 3, num_rows)  # Slope of peak exercise ST segment
ca = np.random.randint(0, 5, num_rows)  # Number of major vessels colored by fluoroscopy
thal = np.random.randint(1, 4, num_rows)  # Thalassemia: 1 = normal, 2 = fixed defect, 3 = reversable defect
target = np.random.choice([0, 1], num_rows)  # 0 = no heart disease, 1 = heart disease

# Create a DataFrame
columns = [
    "age", "sex", "cp", "trestbps", "chol", "fbs",
    "restecg", "thalach", "exang", "oldpeak", "slope",
    "ca", "thal", "target"
]
data = pd.DataFrame({
    "age": age, "sex": sex, "cp": cp, "trestbps": trestbps,
    "chol": chol, "fbs": fbs, "restecg": restecg, "thalach": thalach,
    "exang": exang, "oldpeak": oldpeak, "slope": slope,
    "ca": ca, "thal": thal, "target": target
}, columns=columns)

# Save the dataset to a CSV file
csv_file = "heart_disease.csv"
data.to_csv(csv_file, index=False)
print(f"Synthetic dataset created and saved as {csv_file}")
```

```
Synthetic dataset created and saved as heart_disease.csv
```

In [8]:
```
# Load the dataset
data = pd.read_csv('heart_disease.csv')

# Display basic info about the dataset
print("Dataset Preview:")
print(data.head())

# Summary statistics
print("\nSummary Statistics:")
print(data.describe())
```

```
Dataset Preview:
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   67    1   1       118   448    1        1      128      1      0.0      0
1   57    0   2       164   181    0        0       91      0      3.9      2
2   43    0   2       145   165    1        1      152      0      2.8      0
3   71    0   3       163   479    1        0      139      0      0.8      0
4   36    1   2       181   165    1        0      201      0      5.9      0

   ca  thal  target
0   1     2       0
1   4     2       0
2   0     1       1
3   2     3       0
4   1     2       1

Summary Statistics:
              age         sex          cp    trestbps        chol         fbs  \
count  500.000000  500.000000  500.000000  500.000000  500.000000  500.000000
mean    54.406000    0.480000    1.478000  146.236000  345.740000    0.526000
std     14.608353    0.500100    1.089904   30.591109  129.913492    0.499824
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     43.000000    0.000000    1.000000  119.000000  231.750000    0.000000
50%     55.000000    0.000000    1.000000  147.000000  349.000000    1.000000
75%     67.000000    1.000000    2.000000  174.000000  459.250000    1.000000
max     79.000000    1.000000    3.000000  199.000000  563.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
count  500.000000  500.000000  500.000000  500.000000  500.000000  500.000000
mean     0.960000  138.498000    0.500000    3.086200    1.048000    2.040000
std      0.812231   38.152184    0.500501    1.792332    0.819169    1.420717
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  108.000000    0.000000    1.500000    0.000000    1.000000
50%      1.000000  139.000000    0.500000    3.100000    1.000000    2.000000
75%      2.000000  171.250000    1.000000    4.600000    2.000000    3.000000
max      2.000000  201.000000    1.000000    6.200000    2.000000    4.000000

             thal      target
count  500.000000  500.000000
mean     2.026000    0.460000
std      0.804125    0.498897
min      1.000000    0.000000
25%      1.000000    0.000000
50%      2.000000    0.000000
75%      3.000000    1.000000
max      3.000000    1.000000
```

In [9]:
```
# ---------------------- Handle Missing Values ----------------------
print("\nChecking for missing values...")
print(data.isnull().sum())  # Display missing values in each column

# Fill missing numerical values with the column mean
data.fillna(data.mean(), inplace=True)
print("Missing values handled!")

# ---------------------- Feature and Target Separation ----------------------
# Separate features (X) and target variable (y)
X = data.drop(columns=['target'])  # Replace 'target' with the actual target column name in your dataset
y = data['target']

# ---------------------- Normalize Numerical Features ----------------------
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("\nFeatures normalized successfully!")
```

```
Checking for missing values...
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
Missing values handled!

Features normalized successfully!
```

In [ ]: