In [3]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [4]:
```
df = pd.read_csv("/content/drive/MyDrive/heart disease.zip")
print(df.head())
```

```
   Age  Sex  Chest pain type   BP  Cholesterol  FBS over 120  EKG results  \
0   70    1                4  130          322             0            2
1   67    0                3  115          564             0            2
2   57    1                2  124          261             0            0
3   64    1                4  128          263             0            0
4   74    0                2  120          269             0            2

   Max HR  Exercise angina  ST depression  Slope of ST  \
0     109                0            2.4            2
1     160                0            1.6            2
2     141                0            0.3            1
3     105                1            0.2            2
4     121                1            0.2            1

   Number of vessels fluro  Thallium Heart Disease
0                        3         3      Presence
1                        0         7       Absence
2                        0         7      Presence
3                        1         7       Absence
4                        1         3       Absence
```

In [5]:
```
print(df.isnull().sum())
df = df.dropna()
```

```
Age                       0
Sex                       0
Chest pain type           0
BP                        0
Cholesterol               0
FBS over 120              0
EKG results               0
Max HR                    0
Exercise angina           0
ST depression             0
Slope of ST               0
Number of vessels fluro   0
Thallium                  0
Heart Disease             0
dtype: int64
```

In [7]:
```
X = df.drop("Heart Disease", axis=1)
y = df["Heart Disease"]
```

In [8]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
```

```
)
```

In [9]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [10]:
```
model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[10]:
```
▾ LogisticRegression
LogisticRegression()
```

In [11]:
```
y_pred = model.predict(X_test)
```

In [13]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9074074074074074
```

In [15]:
```
# ----------------------------
# STEP 10 — Predict using USER INPUT
# ----------------------------

print("Enter the values for prediction:")

# Enter values in the same order as the dataset columns
# Example column order:
# age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope,

age = float(input("Enter age: "))
sex = int(input("Enter sex (1=male, 0=female): "))
cp = int(input("Enter chest pain type (0-3): "))
trestbps = float(input("Enter resting blood pressure: "))
chol = float(input("Enter cholesterol: "))
fbs = int(input("Fasting blood sugar > 120 mg/dl (1=yes, 0=no): "))
restecg = int(input("Resting ECG results (0-2): "))
thalach = float(input("Enter maximum heart rate achieved: "))
exang = int(input("Exercise induced angina (1=yes, 0=no): "))
oldpeak = float(input("Enter ST depression value: "))
slope = int(input("Slope of peak exercise ST segment (0-2): "))
ca = int(input("Number of major vessels (0-3): "))
thal = int(input("Enter thal (1,2,3): "))

# Put all values into a list
user_data = [[age, sex, cp, trestbps, chol, fbs, restecg,
              thalach, exang, oldpeak, slope, ca, thal]]

# Scale the input data
user_data = scaler.transform(user_data)

# Make prediction
```

```python
prediction = model.predict(user_data)
probability = model.predict_proba(user_data)[0][1]

# Output
if prediction[0] == 1:
    print("\n🔥 High chance of Heart Disease")
else:
    print("\n✔ Low chance of Heart Disease")

print("Probability of having heart disease:", probability)
```

```
Enter the values for prediction:
Enter age: 58
Enter sex (1=male, 0=female): 1
Enter chest pain type (0-3): 1
Enter resting blood pressure: 130
Enter cholesterol: 240
Fasting blood sugar > 120 mg/dl (1=yes, 0=no): 0
Resting ECG results (0-2): 1
Enter maximum heart rate achieved: 160
Exercise induced angina (1=yes, 0=no): 0
Enter ST depression value: 1.2
Slope of peak exercise ST segment (0-2): 1
Number of major vessels (0-3): 1
Enter thal (1,2,3): 2

✔ Low chance of Heart Disease
Probability of having heart disease: 0.08278678356044823
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserW
arning: X does not have valid feature names, but StandardScaler was fitted with
feature names
  warnings.warn(
```