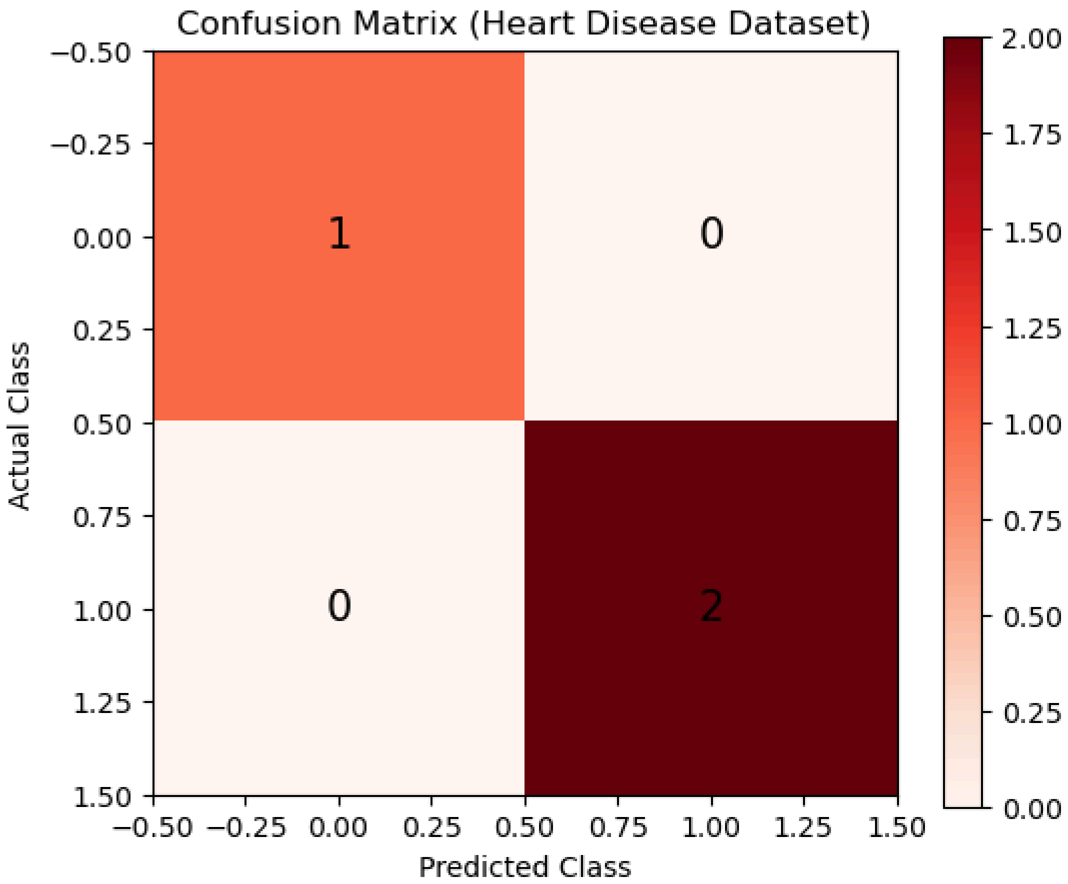
In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, precision_s
df = pd.DataFrame({
    "age":             [52, 46, 57, 62, 48, 59, 67, 51, 54, 60],
    "cholesterol":     [240, 210, 265, 295, 190, 235, 310, 225, 260, 280]
    "max_heart_rate":  [150, 160, 140, 130, 170, 145, 138, 155, 148, 135]
    "blood_pressure":  [130, 120, 140, 150, 118, 136, 145, 128, 138, 142]
    "disease":         [1, 0, 1, 1, 0, 1, 1, 0, 1, 1]   # 1 = Disease, 0
})
X = df[["age", "cholesterol", "max_heart_rate", "blood_pressure"]]
y = df["disease"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=40
)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(cm)
plt.figure(figsize=(6,5))
plt.imshow(cm, cmap="Reds")
plt.title("Confusion Matrix (Heart Disease Dataset)")
plt.xlabel("Predicted Class")
plt.ylabel("Actual Class")
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j], ha="center", va="center", fontsize=15)
plt.colorbar()
plt.show()
accuracy  = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall    = recall_score(y_test, y_pred)
f1        = f1_score(y_test, y_pred)

print("\nAccuracy :", accuracy)
print("Precision:", precision)
print("Recall   :", recall)
print("F1 Score :", f1)
```

```
Confusion Matrix:
[[1 0]
 [0 2]]
```



```
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0
```

In [ ]: