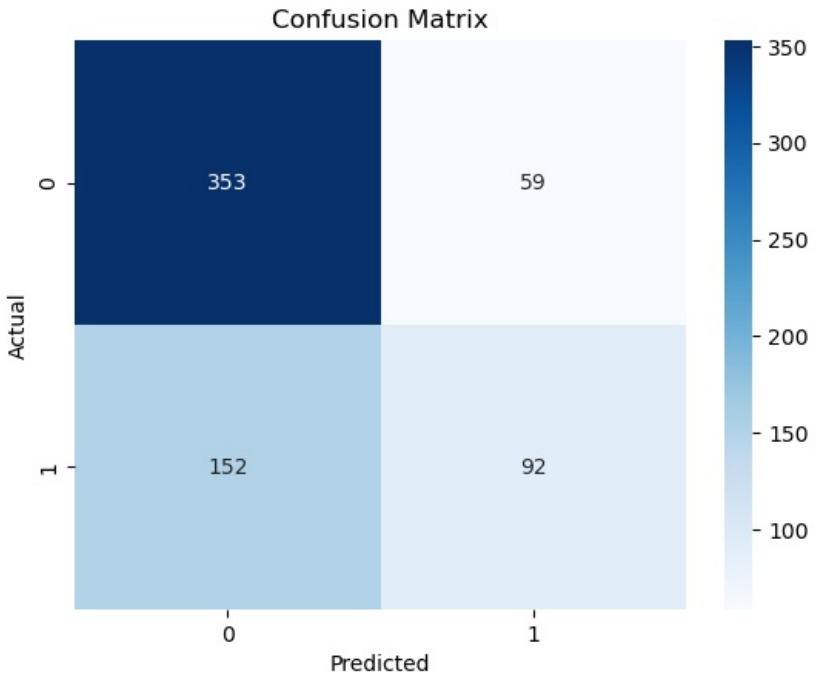
In [2]:
```
!Pip install pandas numpy matplotlib seaborn scikit-learn
```

```
Requirement already satisfied: pandas in c:\users\sidda\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: numpy in c:\users\sidda\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\users\sidda\anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\sidda\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: scikit-learn in c:\users\sidda\anaconda3\lib\site-packages (1.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sidda\anaconda3\lib\site-packages (from pandas
) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\sidda\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\sidda\anaconda3\lib\site-packages (from pandas) (2023.
3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib) (0.1
1.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib)
(4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib) (
24.1)
Requirement already satisfied: pillow>=8 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib) (10.4.0
)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sidda\anaconda3\lib\site-packages (from matplotlib)
(3.1.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\sidda\anaconda3\lib\site-packages (from scikit-learn) (1
.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\sidda\anaconda3\lib\site-packages (from scikit-learn) (
1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\sidda\anaconda3\lib\site-packages (from scikit-l
earn) (3.5.0)
Requirement already satisfied: six>=1.5 in c:\users\sidda\anaconda3\lib\site-packages (from python-dateutil>=2.8
.2->pandas) (1.16.0)
```

In [84]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

# Load the dataset
df = pd.read_csv("C:/Users/sidda/OneDrive/Desktop/KEERTHI/water_potability.csv")

# Display first few rows
print("Dataset Preview:")
print(df.head())

# Check for missing values
print("\nMissing Values:")
print(df.isnull().sum())

# Impute missing values using mean
imputer = SimpleImputer(strategy='mean')
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)

# Split features and labels
X = df_imputed.drop("Potability", axis=1)
y = df_imputed["Potability"]

# Split the dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train a Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train_scaled, y_train)

# Predict on the test set
y_pred = model.predict(X_test_scaled)

# Evaluate the model
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:")
```

```
print(classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
Dataset Preview:
         ph    Hardness        Solids  Chloramines     Sulfate  Conductivity  \
0       NaN  204.890455  20791.318981     7.300212  368.516441    564.308654
1  3.716080  129.422921  18630.057858     6.635246         NaN    592.885359
2  8.099124  224.236259  19909.541732     9.275884         NaN    418.606213
3  8.316766  214.373394  22018.417441     8.059332  356.886136    363.266516
4  9.092223  181.101509  17978.986339     6.546600  310.135738    398.410813

   Organic_carbon  Trihalomethanes  Turbidity  Potability
0       10.379783        86.990970   2.963135           0
1       15.180013        56.329076   4.500656           0
2       16.868637        66.420093   3.055934           0
3       18.436524       100.341674   4.628771           0
4       11.558279        31.997993   4.075075           0

Missing Values:
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64

Accuracy: 0.6783536585365854

Classification Report:
              precision    recall  f1-score   support

         0.0       0.70      0.86      0.77       412
         1.0       0.61      0.38      0.47       244

    accuracy                           0.68       656
   macro avg       0.65      0.62      0.62       656
weighted avg       0.67      0.68      0.66       656
```



In [104…
```
!pip install joblib
```

```
Requirement already satisfied: joblib in c:\users\sidda\anaconda3\lib\site-packages (1.4.2)
```

In [106…
```
import joblib
```

```
# Load the saved scaler
scaler = joblib.load('scaler.pkl')
```

In [116…

```
# train_first.py
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib

# Load your dataset
df = pd.read_csv('C:/Users/sidda/OneDrive/Desktop/KEERTHI/water_potability.csv')  # Make sure this file exists

# Prepare data (adjust column names as needed)
X = df.drop('Potability', axis=1)
y = df['Potability']

# Split and scale data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)

# Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Save model and scaler
joblib.dump(model, 'water_quality_model.pkl')
joblib.dump(scaler, 'scaler.pkl')

print("Model trained and saved successfully!")
print(f"Model accuracy: {model.score(scaler.transform(X_test), y_test):.1%}")
```

```
Model trained and saved successfully!
Model accuracy: 69.1%
```

In [118…

```
# predict_after_training.py
import joblib
import numpy as np

# Load the saved model and scaler
model = joblib.load('water_quality_model.pkl')
scaler = joblib.load('scaler.pkl')

# Get input from user
print("Enter water quality measurements:")
inputs = [
    float(input("pH (0-14): ")),
    float(input("Hardness: ")),
    float(input("Solids (ppm): ")),
    float(input("Chloramines (ppm): ")),
    float(input("Sulfate (mg/L): ")),
    float(input("Conductivity (µS/cm): ")),
    float(input("Organic Carbon (ppm): ")),
    float(input("Trihalomethanes (µg/L): ")),
    float(input("Turbidity (NTU): "))
]

# Make prediction
scaled_input = scaler.transform([inputs])
prediction = model.predict(scaled_input)[0]
probability = model.predict_proba(scaled_input)[0][1]

# Show result
print(f"\nPrediction: {'POTABLE' if prediction == 1 else 'NOT POTABLE'}")
print(f"Confidence: {probability:.1%}")
```

```
Enter water quality measurements:
Prediction: POTABLE
Confidence: 65.0%
C:\Users\sidda\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names
, but StandardScaler was fitted with feature names
  warnings.warn(
```

In [ ]: