```python
# Import required libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

# Load or simulate patient data
# Replace this with real healthcare dataset (e.g., from UCI or Kaggle)
data = pd.DataFrame({
    'age': [25, 45, 52, 36, 23, 50, 40, 60],
    'sex': [1, 0, 1, 0, 1, 1, 0, 0],
    'blood_pressure': [120, 140, 130, 135, 110, 145, 128, 150],
    'cholesterol': [200, 250, 240, 230, 180, 270, 220, 260],
    'diabetes': [0, 1, 1, 0, 0, 1, 0, 1],  # Label: 1 = Disease present
})

# Features and target
X = data.drop('diabetes', axis=1)
y = data['diabetes']

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Example: Predict on new patient
new_patient = pd.DataFrame({
    'age': [34],
    'sex': [1],
    'blood_pressure': [138],
    'cholesterol': [245]
})

prediction = model.predict(new_patient)
print("Predicted disease status (1 = disease, 0 = healthy):", prediction[0])
```

```
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           1       1.00      1.00      1.00         2

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2

Predicted disease status (1 = disease, 0 = healthy): 0
```

Start coding or generate with AI.