

In [34]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree # Added plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
url = '/home/admin1/Downloads/forest.csv'
data = pd.read_csv(url)

# Display the first few rows of the dataset
print(data.head())

# Define features (X) and target variable (y)
X = data.drop('Outcome', axis=1)
y = data['Outcome']

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

# Build a Decision Tree model
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)

# Make predictions on the test set
dt_predictions = dt_model.predict(X_test)

# Evaluate Decision Tree performance
dt_accuracy = accuracy_score(y_test, dt_predictions)
print("Decision Tree Classifier:")
print("Accuracy:", dt_accuracy)
print("Classification Report:")
print(classification_report(y_test, dt_predictions))

# Plot the confusion matrix for Decision Tree
dt_cm = confusion_matrix(y_test, dt_predictions)
plt.figure(figsize=(5, 4))
sns.heatmap(dt_cm, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.title("Decision Tree Confusion Matrix")
plt.show()

# Plot the Decision Tree
plt.figure(figsize=(15, 10))
plot_tree(dt_model, feature_names=X.columns, class_names=["0", "1"], filled=True)
plt.title("Decision Tree Visualization")
plt.show()

# Build a Random Forest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions on the test set
rf_predictions = rf_model.predict(X_test)

# Evaluate Random Forest performance
rf_accuracy = accuracy_score(y_test, rf_predictions)
print("\nRandom Forest Classifier:")
print("Accuracy:", rf_accuracy)
print("Classification Report:")
print(classification_report(y_test, rf_predictions))
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

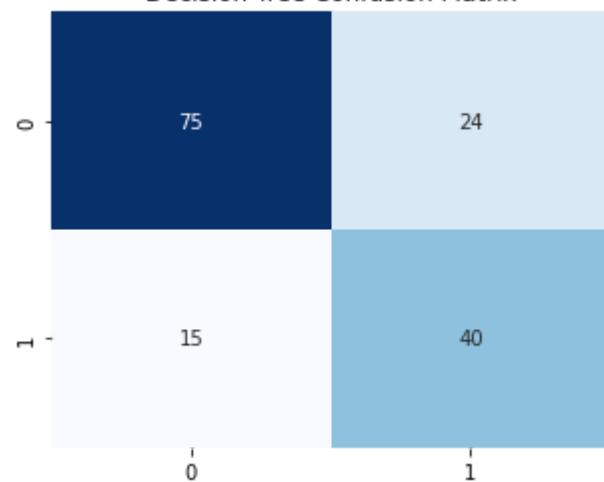
Decision Tree Classifier:

Accuracy: 0.7467532467532467

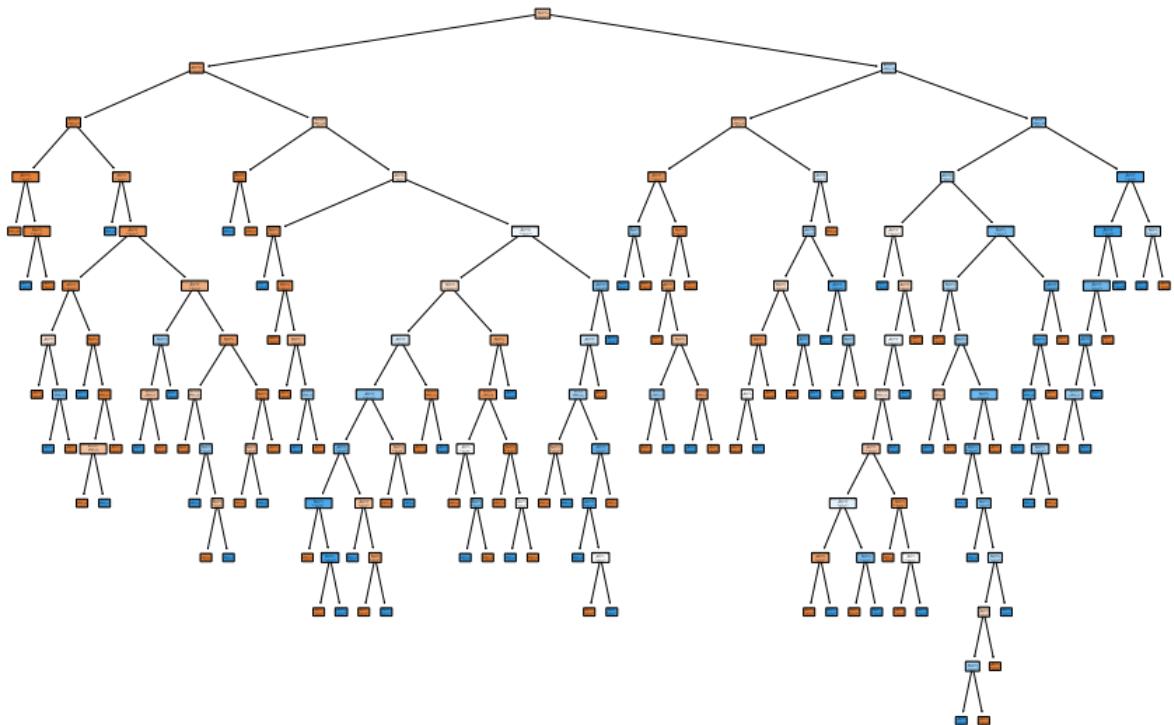
Classification Report:

	precision	recall	f1-score	support
0	0.83	0.76	0.79	99
1	0.62	0.73	0.67	55
accuracy			0.75	154
macro avg	0.73	0.74	0.73	154
weighted avg	0.76	0.75	0.75	154

Decision Tree Confusion Matrix



Decision Tree Visualization



Random Forest Classifier:

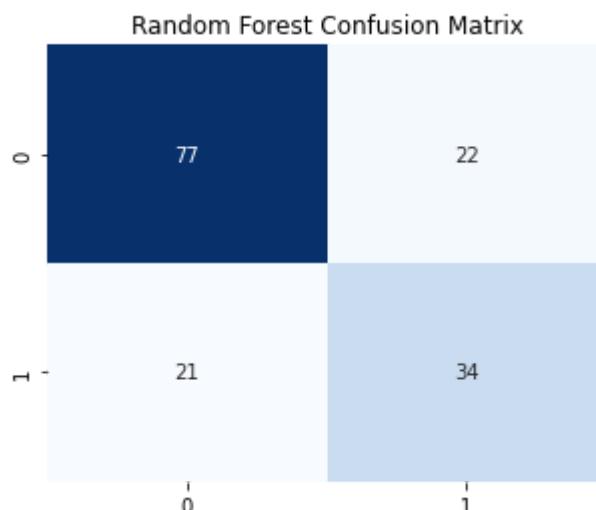
Accuracy: 0.7207792207792207

Classification Report:

	precision	recall	f1-score	support
0	0.79	0.78	0.78	99
1	0.61	0.62	0.61	55
accuracy			0.72	154
macro avg	0.70	0.70	0.70	154
weighted avg	0.72	0.72	0.72	154

In [35]:

```
# Plot the confusion matrix for Random Forest
rf_cm = confusion_matrix(y_test, rf_predictions)
plt.figure(figsize=(5, 4))
sns.heatmap(rf_cm, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.title("Random Forest Confusion Matrix")
plt.show()
```



In []: