In [3]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import graphviz
from sklearn.tree import export_graphviz
```

In [4]:

```
# Load the dataset
df = pd.read_csv('pima-indians-diabetes.csv')

# Display the first 5 records
print(df.head())
```

```
   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
```

In [5]:

```
# Features (X) and target variable (y)
X = df.drop('Outcome', axis=1)
y = df['Outcome']
```

In [6]:

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

In [7]:

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

In [8]:

```
# Create a decision tree classifier
clf = DecisionTreeClassifier(random_state=42)

# Train the classifier
clf.fit(X_train, y_train)
```

Out[8]:

```
▾        DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

In [9]:

```
# Make predictions
y_pred = clf.predict(X_test)
```

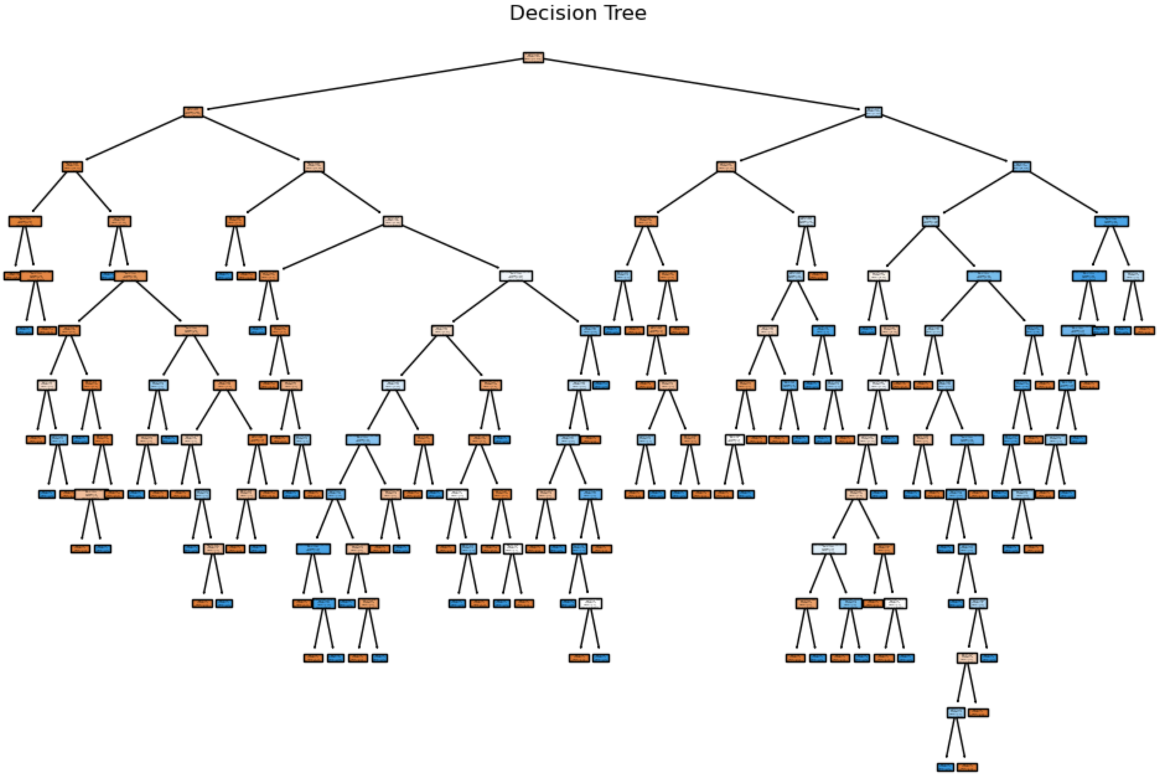
In [10]:

```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
```

```
Model Accuracy: 0.75
```

In [11]:

```
# Plot the decision tree
plt.figure(figsize=(12, 8))
plot_tree(clf, filled=True, feature_names=X.columns, class_names=['No Diabetes', 'Diabet
plt.title("Decision Tree")
plt.show()
```



In [12]:

```
# Export the tree to a .dot file
dot_data = export_graphviz(clf, out_file=None,
                           feature_names=X.columns,
                           class_names=['No Diabetes', 'Diabetes'],
                           filled=True, rounded=True,
                           special_characters=True)

# Create a graph from the dot data
graph = graphviz.Source(dot_data)
graph.render("diabetes_tree")  # Save the tree as a PDF
graph.view()  # Display the tree
```

Out[12]:

```
'diabetes_tree.pdf'
```

In [ ]:

```

```