Démonstration de l'Arbre de décision en utilisant la base IRIS

Importation des Bibliothèques

In [1]:
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
import matplotlib.pyplot as plt
```

Charger le dataset IRIS

In [2]:
```
iris = datasets.load_iris()
```

Normalisation des données

In [ ]:
```
X = iris.data  # Features (sepal length, sepal width, petal length, petal width)
Y = iris.target  # Target labels (species)
```

Séparer en données d'entraînement et de test

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_
```

Entraîner un classifieur DecisionTreeClassifier

Le paramètre `random_state` contrôle l'aléatoire de l'estimateur.

In [ ]:
```
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```

Out[ ]:
```
DecisionTreeClassifier(random_state=42)
```

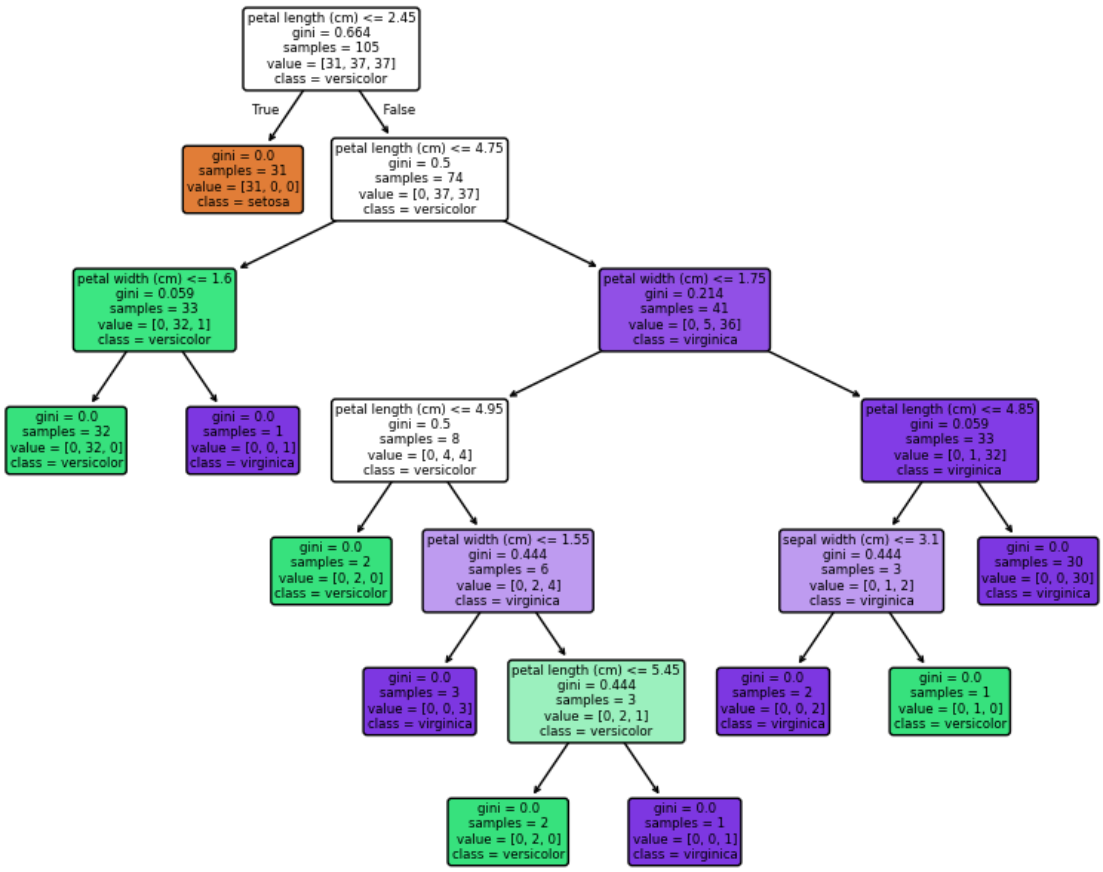
Prédictions et évaluation

In [6]:
```
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of the Decision Tree Classifier: {accuracy * 100:.2f}%")
```

```
Accuracy of the Decision Tree Classifier: 100.00%
```

Visualisation de l'arbre de décision

In [7]:
```
plt.figure(figsize=(10, 8))
tree.plot_tree(clf, filled=True, feature_names=iris.feature_names, class_names=i
plt.show()
```



Test de classification avec un exemple

Création d'un échantillon

In [8]:
```
# Define a new sample (e.g., new iris flower with specific features)
new_sample = [[5.5, 2.4, 3.8, 1.1]]  # Example: sepal length = 5.5, sepal width
```

Prédiction de la classe de note échantillon et affichage

In [9]:
```
# Predict the class for the new sample
predicted_class = clf.predict(new_sample)
print(f"The predicted class for the new sample is: {iris.target_names[predicted_
```

```
The predicted class for the new sample is: versicolor
```