In [1]:
```
#Name: Twinkal S. Bandwal
#Intern at LGM
#Topic:Prediction using Decision Tree Algorithm
```

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

In [2]:
```
#import dataset
data=load_iris()
df=pd.DataFrame(data.data,columns=data.feature_names)
df['target']=data.target
df.head()
```

Out[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [3]:
```
#checking for null values
df.isnull().sum()
```

Out[3]:
```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
dtype: int64
```

In [4]:
```
df.shape
```

Out[4]:
```
(150, 5)
```

In [5]:
```
print(df['target'])
```

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: target, Length: 150, dtype: int32
```

In [6]:
```
#spliting data
fc=[x for x in df.columns if x!="target"]
x=df[fc]
y=df["target"]
X_train ,X_test,Y_train,Y_test=train_test_split(x,y,random_state=100,test_size=0.
```

In [7]:
```
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(105, 4)
(45, 4)
(105,)
(45,)
```

In [8]:
```
modell=DecisionTreeClassifier()
```

In [9]:
```
modell.fit(X_train,Y_train)
```

Out[9]:
```
DecisionTreeClassifier()
```

In [10]:
```
Y_pred=modell.predict(X_test)
```

In [11]:
```
data2=pd.DataFrame({"Actual":Y_test,"Predicted":Y_pred})
data2.head()
```

Out[11]:

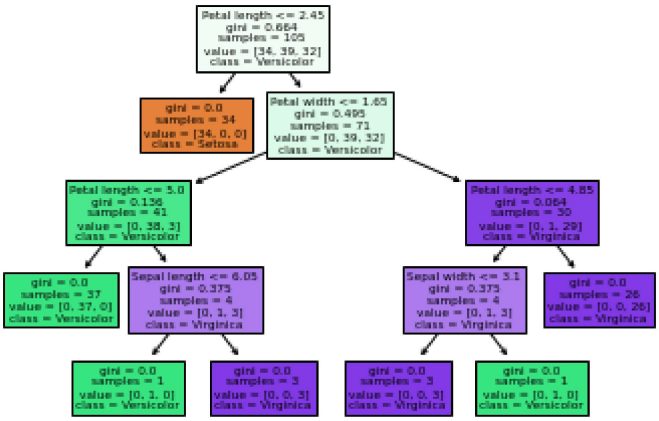
	Actual	Predicted
128	2	2
11	0	0
118	2	2
15	0	0
123	2	2

In [12]:
```
accuracy_score(Y_test,Y_pred)
```

Out[12]:
```
0.9555555555555556
```

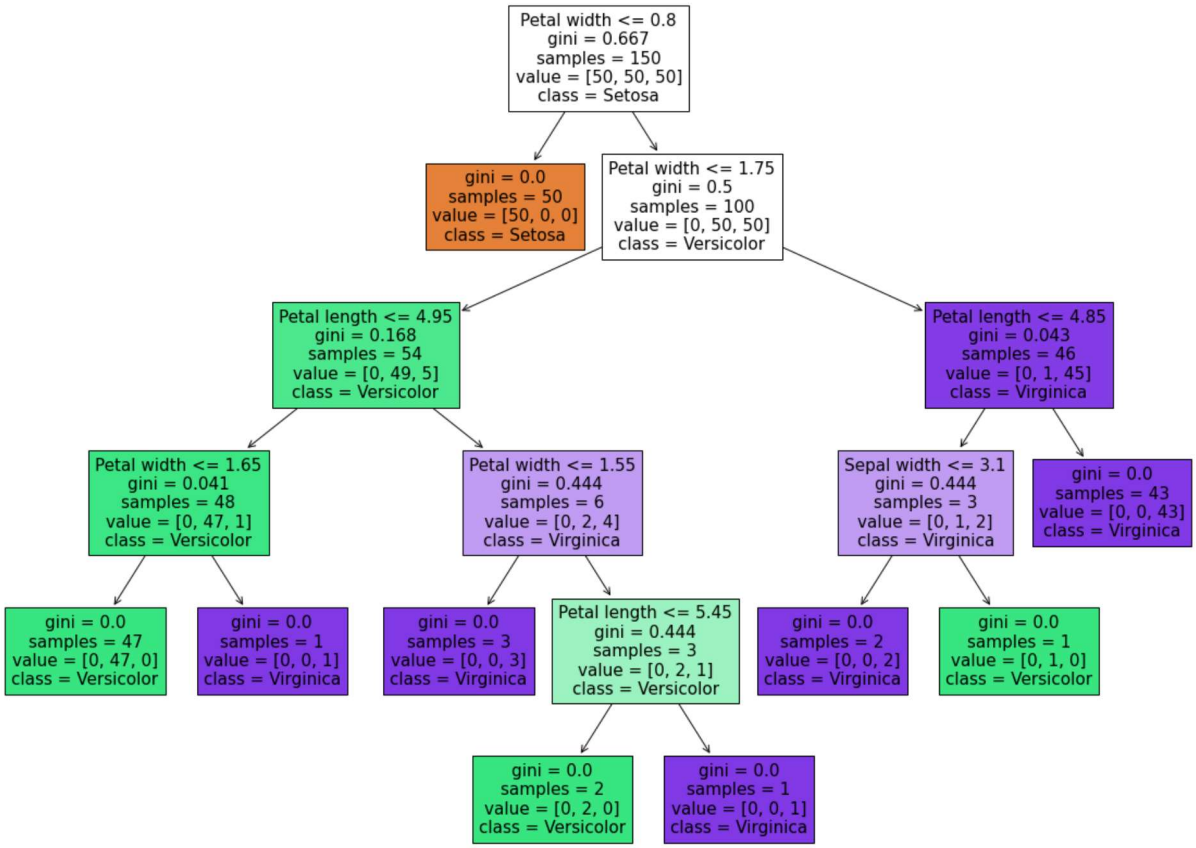
In [13]:
```
#plotting
f_n=["Sepal length", "Sepal width", "Petal length", "Petal width"]
c_n=["Setosa","Versicolor","Virginica"]
plot_tree(modell,feature_names=f_n,class_names=c_n,filled=True)
```

Out[13]:
```
[Text(133.92000000000002, 195.696, 'Petal length <= 2.45\ngini = 0.664\nsamples
= 105\nvalue = [34, 39, 32]\nclass = Versicolor'),
 Text(100.44000000000001, 152.208, 'gini = 0.0\nsamples = 34\nvalue = [34, 0,
0]\nclass = Setosa'),
 Text(167.40000000000003, 152.208, 'Petal width <= 1.65\ngini = 0.495\nsamples
= 71\nvalue = [0, 39, 32]\nclass = Versicolor'),
 Text(66.96000000000001, 108.72, 'Petal length <= 5.0\ngini = 0.136\nsamples =
41\nvalue = [0, 38, 3]\nclass = Versicolor'),
 Text(33.480000000000004, 65.232, 'gini = 0.0\nsamples = 37\nvalue = [0, 37, 0]
\nclass = Versicolor'),
 Text(100.44000000000001, 65.232, 'Sepal length <= 6.05\ngini = 0.375\nsamples
= 4\nvalue = [0, 1, 3]\nclass = Virginica'),
 Text(66.96000000000001, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nc
lass = Versicolor'),
 Text(133.92000000000002, 21.744, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\n
class = Virginica'),
 Text(267.84000000000003, 108.72, 'Petal length <= 4.85\ngini = 0.064\nsamples
= 30\nvalue = [0, 1, 29]\nclass = Virginica'),
 Text(234.36, 65.232, 'Sepal width <= 3.1\ngini = 0.375\nsamples = 4\nvalue =
[0, 1, 3]\nclass = Virginica'),
 Text(200.88000000000002, 21.744, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\n
class = Virginica'),
 Text(267.84000000000003, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]\n
class = Versicolor'),
 Text(301.32000000000005, 65.232, 'gini = 0.0\nsamples = 26\nvalue = [0, 0, 26]
\nclass = Virginica')]
```



In [14]:
```
modelx=DecisionTreeClassifier().fit(x,y)
```

In [15]:
```
plt.figure(figsize=(20,15))
tree=tree.plot_tree(modelx,feature_names=f_n,class_names=c_n,filled=True)
```



In [ ]:
```
```