Task 3 : Prediction using decision tree algorithm

Name: Sharad Parmar

objective: In this task, we shall create the Decision Tree Classifier and visualize it graphically

In [13]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.tree import DecisionTreeRegressor
```

In [4]:
```
df= pd.read_csv("Iris.csv")
```

In [5]:
```
df.head()
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Information and statistics of datasets

In [6]:
```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 6.5+ KB
```

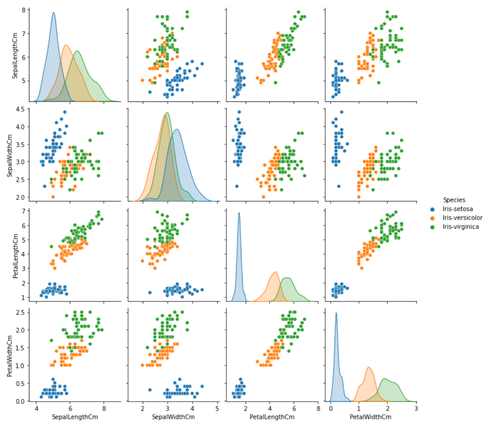
Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [8]:
```
X= df.iloc [:,1:-1]
Y= df.iloc[:,-1]
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.1,random_state=0)
```

In [9]:
```
df=df.iloc[:,1:]
sns.pairplot(df,hue='Species')
```

Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x1b9e8d78>
```



In [10]:
```
lm= DecisionTreeClassifier(random_state=0)
lm.fit(X_train,Y_train)
```

Out[10]:
```
DecisionTreeClassifier(random_state=0)
```

In [11]:
```
Y_train_pred=lm.predict(X_train)
Y_test_pred=lm.predict(X_test)
print("Train accuracy score:",accuracy_score(Y_train,Y_train_pred))
print("Test accuracy score:",accuracy_score(Y_test,Y_test_pred))
```

```
Train accuracy score: 1.0
Test accuracy score: 1.0
```

In [12]:
```
plt.figure (figsize=(15,15))
plot_tree(lm,filled= True)
print()
```