In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
data=pd.read_csv('Iris.csv.csv')
data
```

Out[2]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [3]:

```
data=data.drop('Id',axis=1)
```

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [5]:

```
data.describe()
```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [8]:

```
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

In [9]:

```
mod=DecisionTreeClassifier(criterion='entropy')
```

In [10]:

```
x=data.drop('Species',axis=1)
y=data['Species']
```

In [11]:

```
X_train, X_test, Y_train, Y_test = train_test_split(x,y,test_size=0.25, random_state=0)
```

In [12]:

```
X_train.shape
```

Out[12]:

```
(112, 4)
```

In [13]:

```
Y_train.shape
```

Out[13]:

```
(112,)
```

In [14]:

```
model1=mod.fit(X_train,Y_train)
```

In [15]:

```
y_pred=mod.predict(X_test)
```

In [16]:

```
print(classification_report(y_pred,Y_test))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        13
Iris-versicolor       0.94      1.00      0.97        15
 Iris-virginica       1.00      0.90      0.95        10

       accuracy                           0.97        38
```

```
      macro avg       0.98      0.97      0.97        38
   weighted avg       0.98      0.97      0.97        38
```

In [17]:

```
accuracy_score(y_pred,Y_test)
```

Out[17]:

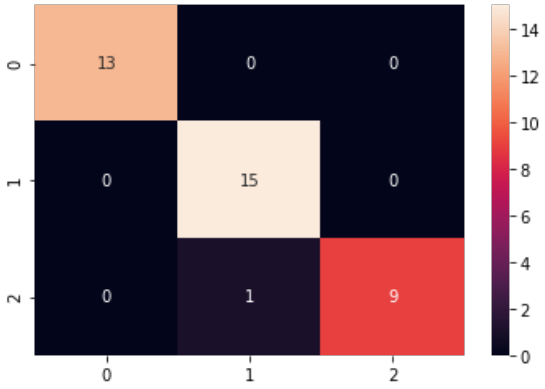
```
0.9736842105263158
```

In [18]:

```
sns.heatmap(confusion_matrix(y_pred,Y_test),annot=True)
```

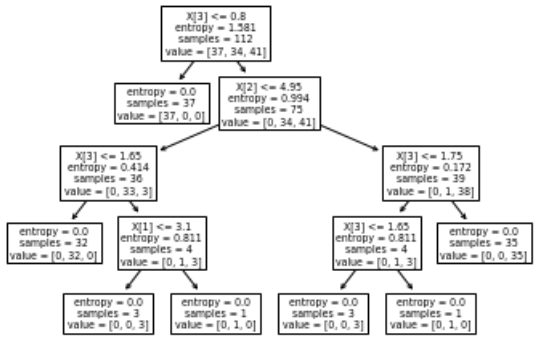
Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a26c349d48>
```



In [19]:

```
from sklearn import tree
tree.plot_tree(mod);
```



In [20]:

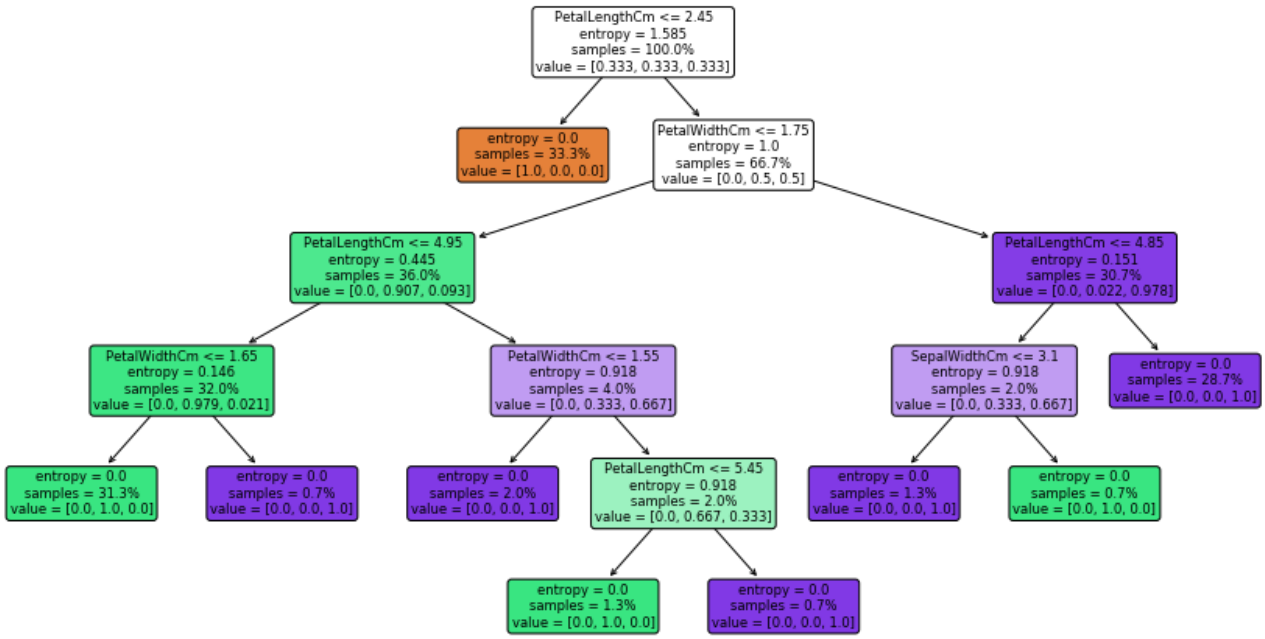
```
cols=list(data.columns.values)
cols
```

Out[20]:

```
['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']
```

In [21]:

```
plt.figure(figsize=(15,8))
tree.plot_tree(mod.fit(x,y),feature_names=cols,filled=True,precision=3,
                                                        proportion=True,rounded=True)
plt.show();
```


PetalLengthCm <= 2.45
entropy = 1.585
samples = 100.0%
value = [0.333, 0.333, 0.333]
entropy = 0.0
samples = 33.3%
value = [1.0, 0.0, 0.0]
PetalWidthCm <= 1.75
entropy = 1.0
samples = 66.7%
value = [0.0, 0.5, 0.5]
PetalLengthCm <= 4.95
entropy = 0.445
samples = 36.0%
value = [0.0, 0.907, 0.093]
PetalLengthCm <= 4.85
entropy = 0.151
samples = 30.7%
value = [0.0, 0.022, 0.978]
PetalWidthCm <= 1.65
entropy = 0.146
samples = 32.0%
value = [0.0, 0.979, 0.021]
PetalWidthCm <= 1.55
entropy = 0.918
samples = 4.0%
value = [0.0, 0.333, 0.667]
SepalWidthCm <= 3.1
entropy = 0.918
samples = 2.0%
value = [0.0, 0.333, 0.667]
entropy = 0.0
samples = 28.7%
value = [0.0, 0.0, 1.0]
entropy = 0.0
samples = 31.3%
value = [0.0, 1.0, 0.0]
entropy = 0.0
samples = 0.7%
value = [0.0, 0.0, 1.0]
entropy = 0.0
samples = 2.0%
value = [0.0, 0.0, 1.0]
PetalLengthCm <= 5.45
entropy = 0.918
samples = 2.0%
value = [0.0, 0.667, 0.333]
entropy = 0.0
samples = 1.3%
value = [0.0, 1.0, 0.0]
entropy = 0.0
samples = 0.7%
value = [0.0, 0.0, 1.0]
entropy = 0.0
samples = 1.3%
value = [0.0, 0.0, 1.0]
entropy = 0.0
samples = 0.7%
value = [0.0, 1.0, 0.0]

In [ ]: