In [29]:

```
import pandas as pd
import os

df = pd.read_csv("Data_.csv")
```

In [30]:

```
lf=df
lf
```

Out[30]:

	Signal_Strength	Interference	Error_Rate	Synchronization	Optimization
0	Low	Low	Low	S	No
1	Low	Low	High	O	Yes
2	Low	Medium	Low	S	No
3	Low	Medium	Medium	O	Yes
4	Low	High	High	O	Yes
5	Low	High	High	S	Yes
6	Low	High	Medium	S	Yes
7	Low	High	Low	S	No
8	Low	Medium	High	S	Yes
9	Medium	Low	Medium	S	No
10	Medium	Low	Low	O	Yes
11	Medium	Medium	High	S	Yes
12	Medium	Medium	High	O	Yes
13	Medium	Low	Low	S	No
14	Medium	High	Low	O	Yes
15	Medium	High	Low	S	No
16	Medium	Medium	Medium	S	No
17	Medium	High	High	O	Yes
18	Medium	High	Medium	S	Yes
19	High	Low	Low	O	No
20	High	Low	Low	S	No
21	High	Medium	Medium	S	No
22	High	Medium	High	S	No
23	High	High	High	O	Yes
24	High	High	Medium	S	No
25	High	Low	High	S	No
26	High	Medium	High	O	Yes
27	High	Medium	Low	O	No
28	High	Medium	Low	S	No

In [31]:

```
lf['Signal_Strength']=lf['Signal_Strength'].map({'Low':0,'Medium':1,'High':2})
lf['Interference']= lf['Interference'].map({'Low':0,'Medium':1,'High':2})
lf['Error_Rate']= lf['Error_Rate'].map({'Low':0,'Medium':1,'High':2})
lf['Synchronization']= lf['Synchronization'].map({'O':0,'S':1})
lf['Optimization']= lf['Optimization'].map({'Yes':0,'No':1})
```

In [32]:

```
print (lf)
```

```
    Signal_Strength  Interference  Error_Rate  Synchronization  Optimizati
on
0                  0             0           0                1
1
1                  0             0           2                0
0
2                  0             1           0                1
1
3                  0             1           1                0
0
4                  0             2           2                0
0
5                  0             2           2                1
0
6                  0             2           1                1
0
7                  0             2           0                1
1
8                  0             1           2                1
0
9                  1             0           1                1
1
10                 1             0           0                0
0
11                 1             1           2                1
0
12                 1             1           2                0
0
13                 1             0           0                1
1
14                 1             2           0                0
0
15                 1             2           0                1
1
16                 1             1           1                1
1
17                 1             2           2                0
0
18                 1             2           1                1
0
19                 2             0           0                0
1
20                 2             0           0                1
1
21                 2             1           1                1
1
22                 2             1           2                1
1
23                 2             2           2                0
0
24                 2             2           1                1
1
25                 2             0           2                1
1
26                 2             1           2                0
0
27                 2             1           0                0
1
28                 2             1           0                1
1
```

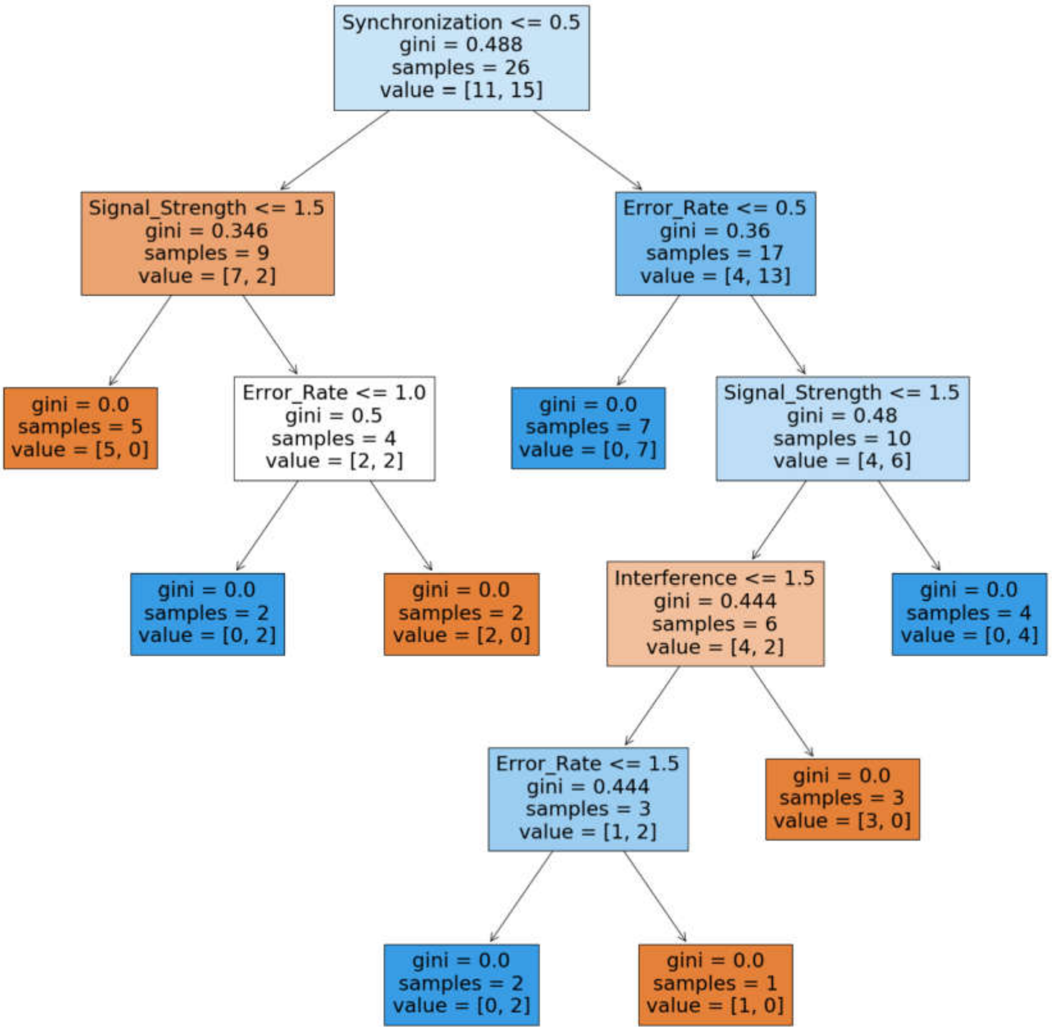
In [36]:

```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
from sklearn.datasets import load_iris
from sklearn import tree
from matplotlib import pyplot as plt
import graphviz
from graphviz import Source

Feature_Columns=['Signal_Strength','Interference','Error_Rate','Synchronization']
X= lf[Feature_Columns]
Y= lf['Optimization']

X_Train, X_Test, Y_Train, Y_Test= train_test_split(X, Y, test_size=0.1)
print("Training Set")
print (X_Train)
print("Testing Set")
print (X_Test)
clf = DecisionTreeClassifier()
train_tree= clf.fit(X_Train,Y_Train)
fig=plt.figure(figsize=(24,24))
tree.plot_tree(clf, feature_names=Feature_Columns, filled=True)
fig.savefig('InitialAttempt.png')
#dot_data = tree.export_graphviz(clf, out_file=None, feature_names=Feature_Columns,filled=True)
#graph = graphviz.Source(dot_data)
#graph.render("Latest",view = True)
y_pred = clf.predict(X_Test)
print("Accuracy:",metrics.accuracy_score(Y_Test, y_pred))
```

```
Training Set
    Signal_Strength  Interference  Error_Rate  Synchronization
24                2             2           1                1
6                 0             2           1                1
0                 0             0           0                1
27                2             1           0                0
11                1             1           2                1
28                2             1           0                1
2                 0             1           0                1
14                1             2           0                0
13                1             0           0                1
10                1             0           0                0
22                2             1           2                1
19                2             0           0                0
26                2             1           2                0
20                2             0           0                1
3                 0             1           1                0
7                 0             2           0                1
17                1             2           2                0
25                2             0           2                1
9                 1             0           1                1
23                2             2           2                0
15                1             2           0                1
5                 0             2           2                1
18                1             2           1                1
16                1             1           1                1
21                2             1           1                1
12                1             1           2                0
Testing Set
   Signal_Strength  Interference  Error_Rate  Synchronization
4                0             2           2                0
1                0             0           2                0
8                0             1           2                1
Accuracy: 1.0
```



In [107]:

```
d = {'Signal_Strength': [0],'Interference': [0], 'Error_Rate': [2], 'Synchronization' :
[0]}
print(clf.predict(pd.DataFrame(data = d)))
```

```
[0]
```

In [ ]:

```

```