In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
def categorize(score):
    if score < 40:
        return 'Low'
    elif score < 70:
        return 'Medium'
    else:
        return 'High'
```

In [4]:

```
data = pd.read_csv("Mall_Customers.csv")
data
```

Out[4]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [5]:

```
data.head()
```

Out[5]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [6]:

```
data.columns = ['CustomerID', 'Gender', 'Age', 'AnnualIncome', 'SpendingScore']
```

In [7]:

```
le = LabelEncoder()
data['Gender'] = le.fit_transform(data['Gender'])  # Male=1, Female=0
```

In [8]:

```
data['SpendingCategory'] = data['SpendingScore'].apply(categorize)
data['SpendingCategory'] = le.fit_transform(data['SpendingCategory'])
```

In [9]:

```
X = data[['Age', 'AnnualIncome', 'Gender']]
y = data['SpendingCategory']
```

In [10]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [11]:

```
dt = DecisionTreeClassifier(max_depth=4, random_state=42)
dt.fit(X_train, y_train)
```

Out[11]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4, random_state=42)
```

In [12]:

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

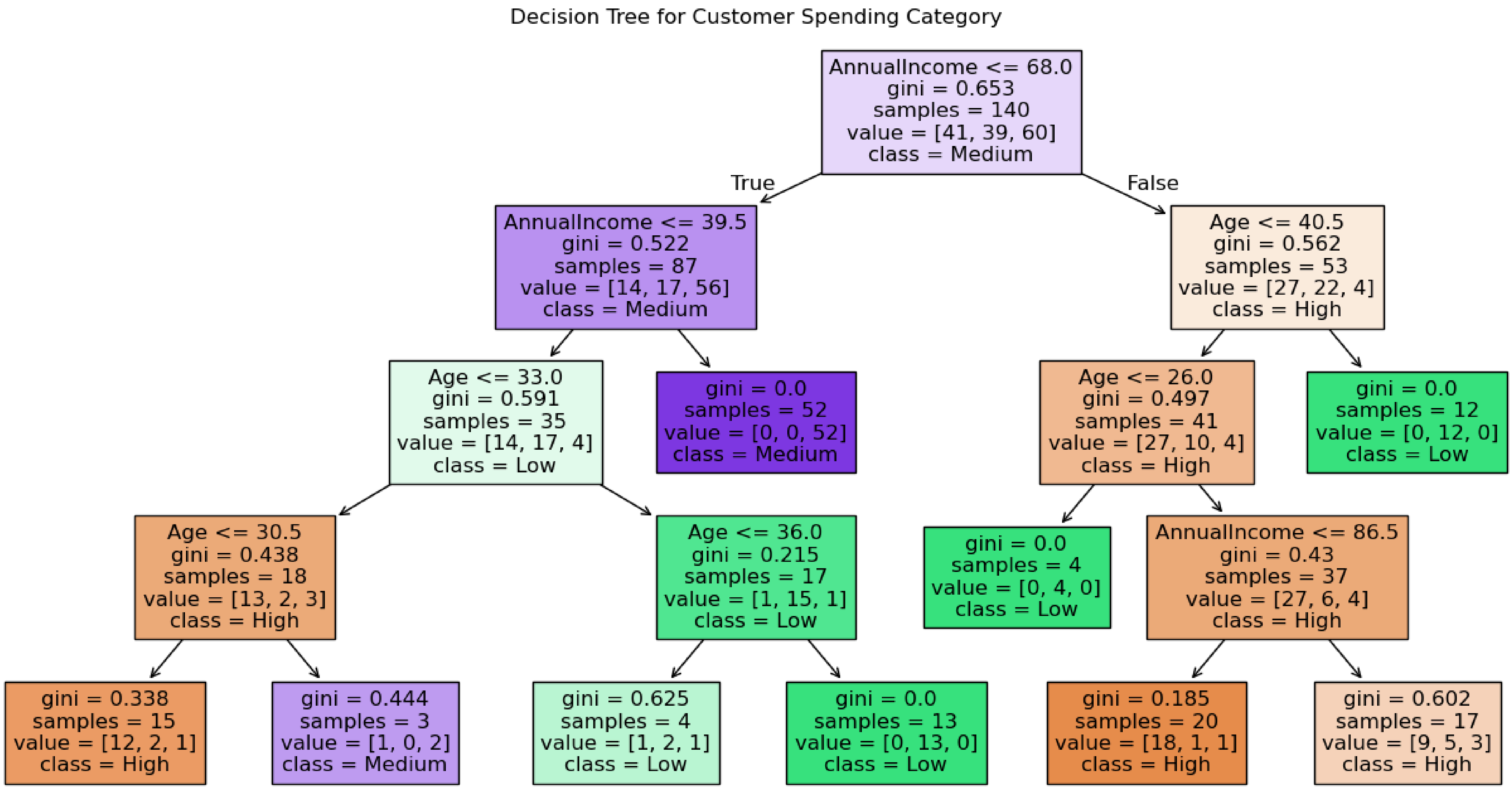
In [13]:

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.8666666666666667

Classification Report:
               precision    recall  f1-score   support

           0       0.63      0.92      0.75        13
           1       0.94      0.80      0.86        20
           2       1.00      0.89      0.94        27

    accuracy                           0.87        60
   macro avg       0.86      0.87      0.85        60
weighted avg       0.90      0.87      0.87        60
```

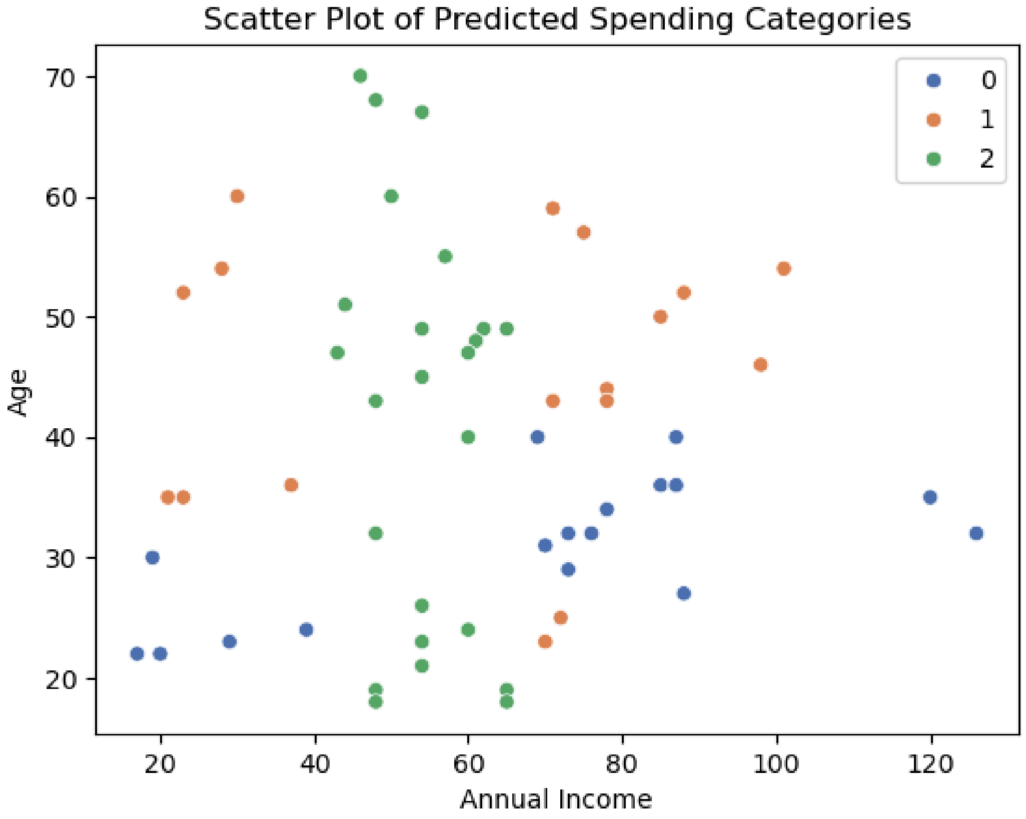
In [14]:

```
plt.figure(figsize=(16, 8))
plot_tree(dt, feature_names=X.columns.tolist(), class_names=['High', 'Low', 'Medium'], filled=True)
plt.title("Decision Tree for Customer Spending Category")
plt.show()
```



In [16]:

```
sns.scatterplot(x=X_test['AnnualIncome'], y=X_test['Age'], hue=y_pred, palette='deep')
plt.title("Scatter Plot of Predicted Spending Categories")
plt.xlabel("Annual Income")
plt.ylabel("Age")
plt.show()
```


Scatter Plot of Predicted Spending Categories
0
1
2
70
60
50
40
30
20
Age
20
40
60
80
100
120
Annual Income

```
In [ ]:
```