In [15]:

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

In [16]:

```
df=pd.read_csv('loan10.csv')
df
```

Out[16]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [17]:

```
df.columns
```

Out[17]:

```
Index(['Home Owner', 'Marital Status', 'Annual Income', 'Defaulted Borrowe
r'], dtype='object')
```

In [18]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Home Owner          10 non-null     object
 1   Marital Status      10 non-null     object
 2   Annual Income       10 non-null     int64 
 3   Defaulted Borrower  10 non-null     object
dtypes: int64(1), object(3)
memory usage: 448.0+ bytes
```

In [20]:

```
df['Defaulted Borrower'].value_counts()
```

Out[20]:

```
No     7
Yes    3
Name: Defaulted Borrower, dtype: int64
```

In [21]:

```
x=df[['Annual Income']]
y=df['Defaulted Borrower']
```

In [22]:

```
d={"Defaulted Borrowe":{'No':1,'Yes':2}}
df=df.replace(df)
print(df)
```

```
  Home Owner Marital Status  Annual Income Defaulted Borrower
0        Yes         Single            125                 No
1         No        Married            100                 No
2         No         Single             70                 No
3        Yes        Married            120                 No
4         No       Divorced             95                Yes
5         No        Married             60                 No
6        Yes       Divorced            220                 No
7         No         Single             85                Yes
8         No        Married             75                 No
9         No         Single             90                Yes
```

In [23]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

In [24]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[24]:

```
RandomForestClassifier()
```

Depth of Tree

In [25]:

```
parameters={"max_depth":[1,2,3,4,5],"min_samples_leaf":[5,23,45,76,78],'n_estimators':[10
```

Cross Validate

In [26]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.
py:666: UserWarning: The least populated class in y has only 1 members, wh
ich is less than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[26]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 23, 45, 76, 78],
                         'n_estimators': [10, 23, 45, 65, 7]},
             scoring='accuracy')
```

Score

In [27]:

```
grid_search.best_score_
```

Out[27]:

```
0.75
```

In [28]:

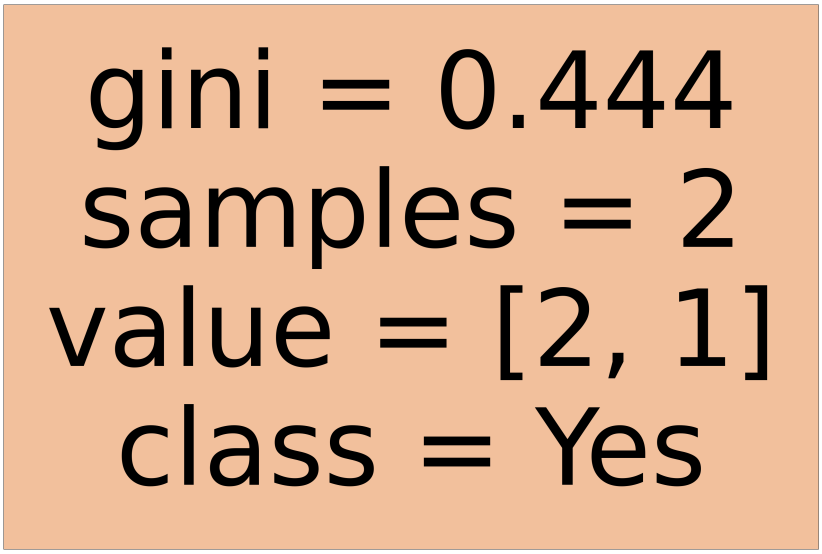
```
rfc_best=grid_search.best_estimator_
```

In [29]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled
```

Out[29]:

```
[Text(2232.0, 1087.2, 'gini = 0.444\nsamples = 2\nvalue = [2, 1]\nclass =
Yes')]
```



In [ ]:

```

```