In [30]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [3]:

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

In [4]:

```
df.head()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [7]:

```
df.info()
```

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

In [8]:

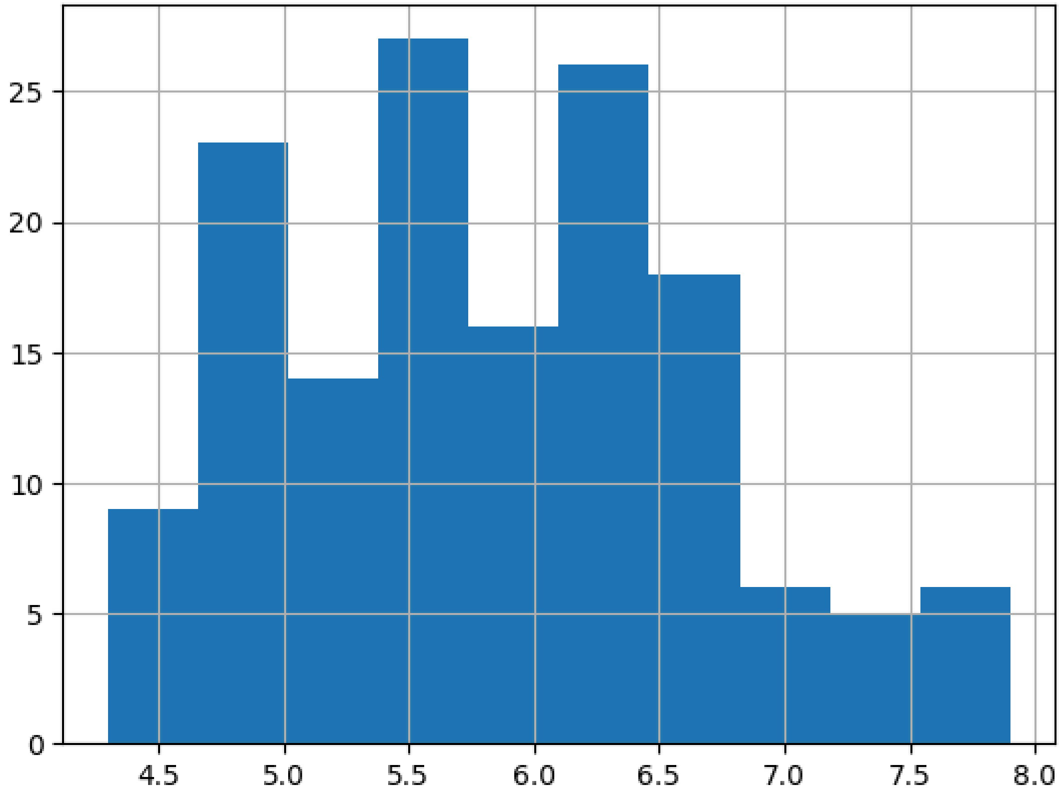
```
df.describe()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
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

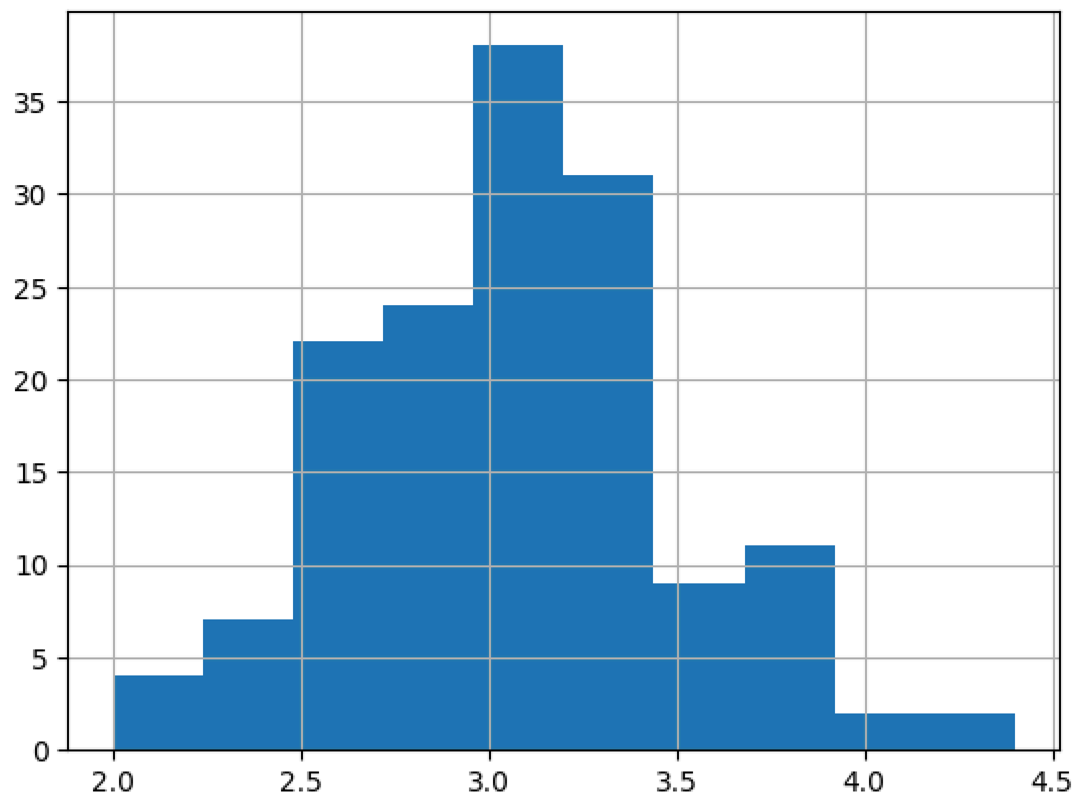
```
In [12]: df['sepal_length'].hist()
```

```
Out[12]: <Axes: >
```



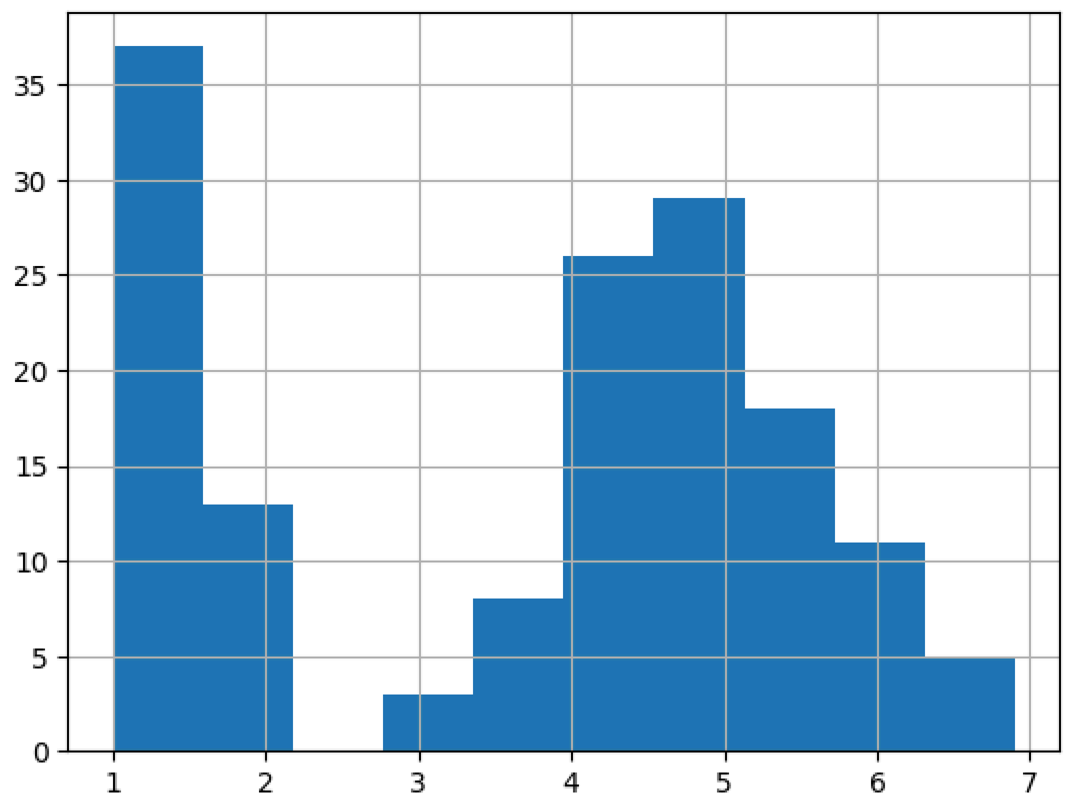
```
In [13]: df['sepal_width'].hist()
```

```
Out[13]: <Axes: >
```



```
In [25]: df['petal_length'].hist()
```

```
Out[25]: <Axes: >
```

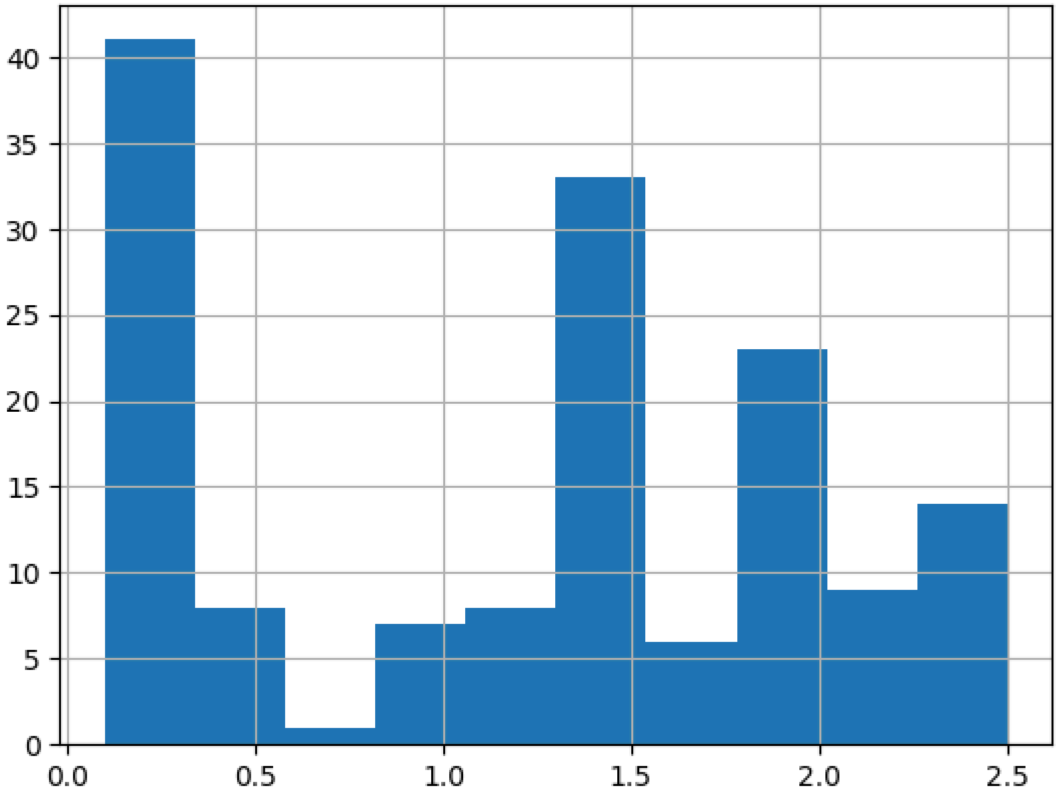


In [26]:

```
df['petal_width'].hist()
```

Out[26]:

```
<Axes: >
```

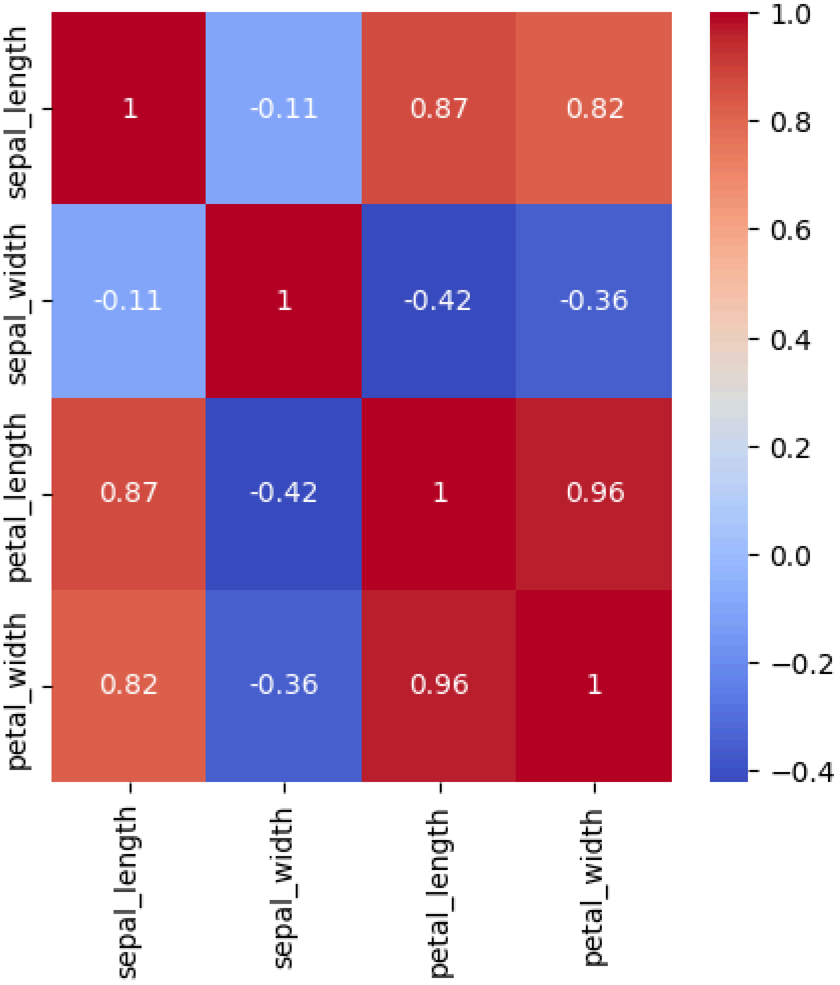


In [24]:

```
numeric_columns = df.drop(columns='species')
corr=numeric_columns.corr()
fig,axis=plt.subplots(figsize=(5,5))
sns.heatmap(corr,annot=True,ax=axis,cmap='coolwarm')
```

Out[24]:

```
<Axes: >
```



```
In [28]: x=df.drop(columns='species')
         y=df['species']
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
In [31]: LR=LogisticRegression()
         LR.fit(x_train,y_train)
```

Out[31]:

▾ LogisticRegression

Parameters

	penalty	'l2'
	dual	False
	tol	0.0001
	C	1.0
	fit_intercept	True
	intercept_scaling	1
	class_weight	None
	random_state	None
	solver	'lbfgs'
	max_iter	100
	multi_class	'deprecated'
	verbose	0
	warm_start	False
	n_jobs	None
	l1_ratio	None

In [32]:

```
LR_accuracy=LR.score(x_test,y_test)
```

In [34]:

```
print(f"Accuracy by using Logistic Regression: {LR_accuracy*100}%")
```

```
Accuracy by using Logistic Regression: 100.0%
```

In [ ]: