In [12]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

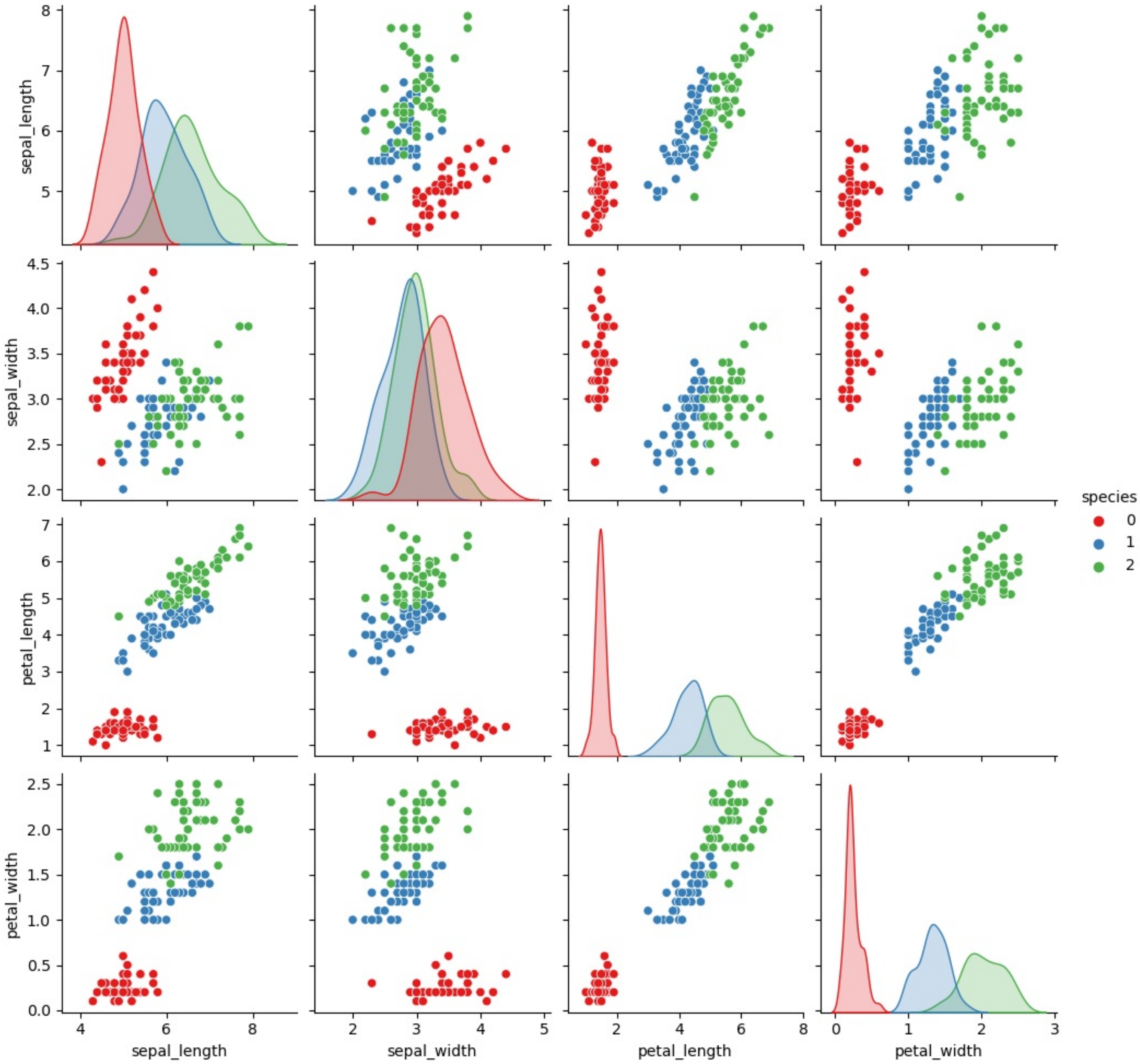
In [4]:
```python
# loading the Dataset
IRIS_data = pd.read_csv(r'C:\Users\admin\Desktop\2213557-MA 336\IRIS.csv')
```

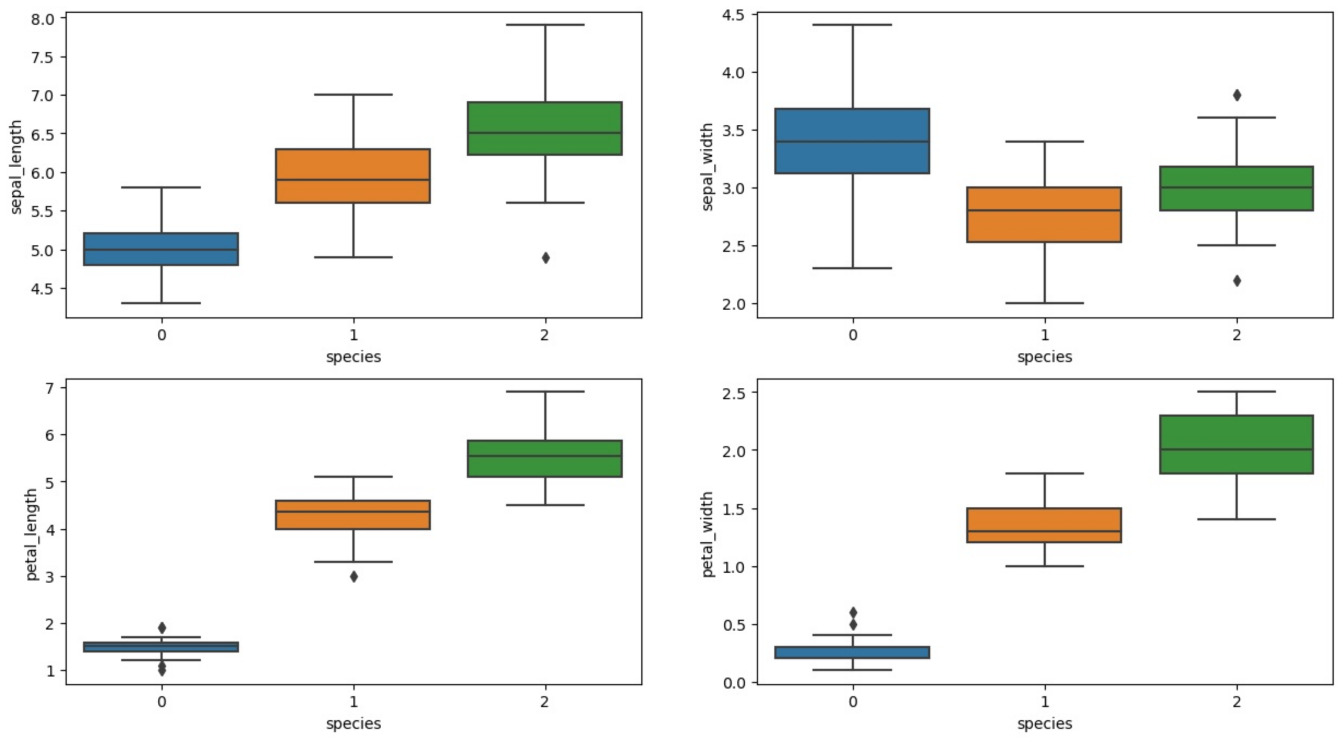
Exploring the Data Set

In [5]:
```python
# Display the first few rows of the dataset
print(IRIS_data.head())

# Check for missing values
print(IRIS_data.isnull().sum())

# Explore the distribution of classes
print(IRIS_data['species'].value_counts())
```

```
   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
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: species, dtype: int64
```

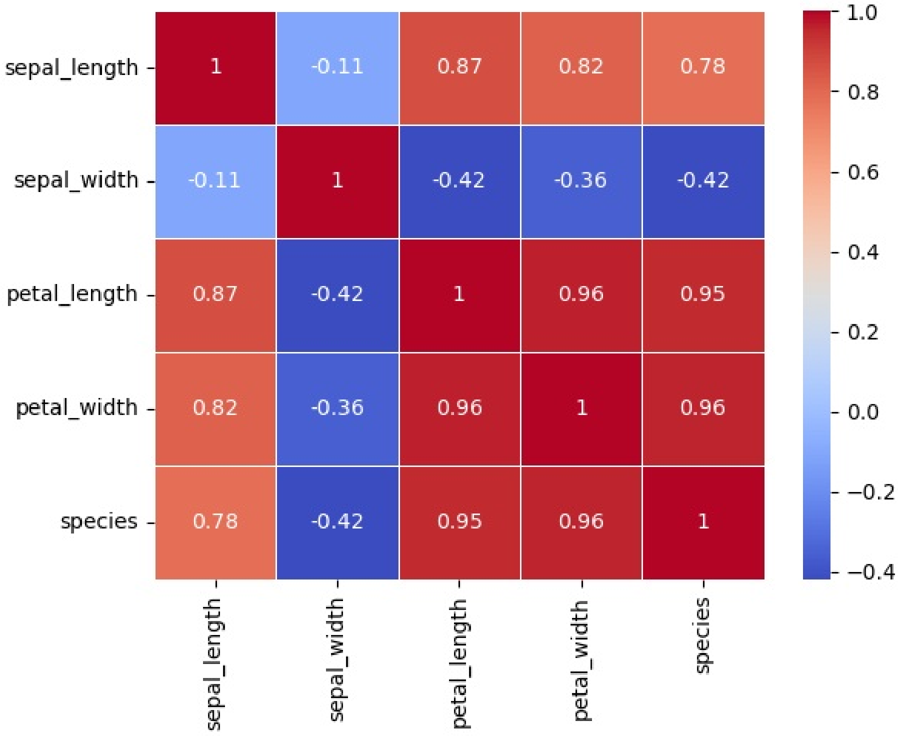
In [14]:
```python
# Pairplot to visualize relationships between features
sns.pairplot(IRIS_data, hue='species', markers='o', palette='Set1')
plt.show()

# Boxplot for each feature by species
plt.figure(figsize=(15, 8))
plt.subplot(2, 2, 1)
sns.boxplot(x='species', y='sepal_length', data=IRIS_data)
plt.subplot(2, 2, 2)
sns.boxplot(x='species', y='sepal_width', data=IRIS_data)
plt.subplot(2, 2, 3)
sns.boxplot(x='species', y='petal_length', data=IRIS_data)
plt.subplot(2, 2, 4)
sns.boxplot(x='species', y='petal_width', data=IRIS_data)
plt.show()
```





In [15]:

```python
# Correlation matrix to check for feature relationships
correlation_matrix = IRIS_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
plt.show()
```



Preprocessing the Data Set

In [6]:
```
# Encode the target variable
le = LabelEncoder()
IRIS_data['species'] = le.fit_transform(IRIS_data['species'])

# Split the data into features (X) and target (y)
X = IRIS_data.drop('species', axis=1)
y = IRIS_data['species']
```

Encoding the categorical target variable ('species') into numerical values and split the dataset into features (X) and target (y)

Split the Dataset:

Spliting the dataset into training and testing sets.

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Train a Machine Learning Model:

Using Random Forest Classifier, to train on the training data.

In [8]:
```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[8]:
```
RandomForestClassifier(random_state=42)
```

Making Predictions

In [9]:
```
y_pred = model.predict(X_test)
```

Evaluating the Model

In [10]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

classification_rep = classification_report(y_test, y_pred)
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 1.00
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

Here,The reported accuracy of 1.00 (100%) signifies exceptional performance by the machine learning model on the test dataset, demonstrating precise classification of Iris flowers into their respective species. The precision, recall, and F1-score metrics, all with perfect scores of 1.00 for each class (species 0, 1, and 2), highlight the model's accuracy and reliability in both positive and negative predictions