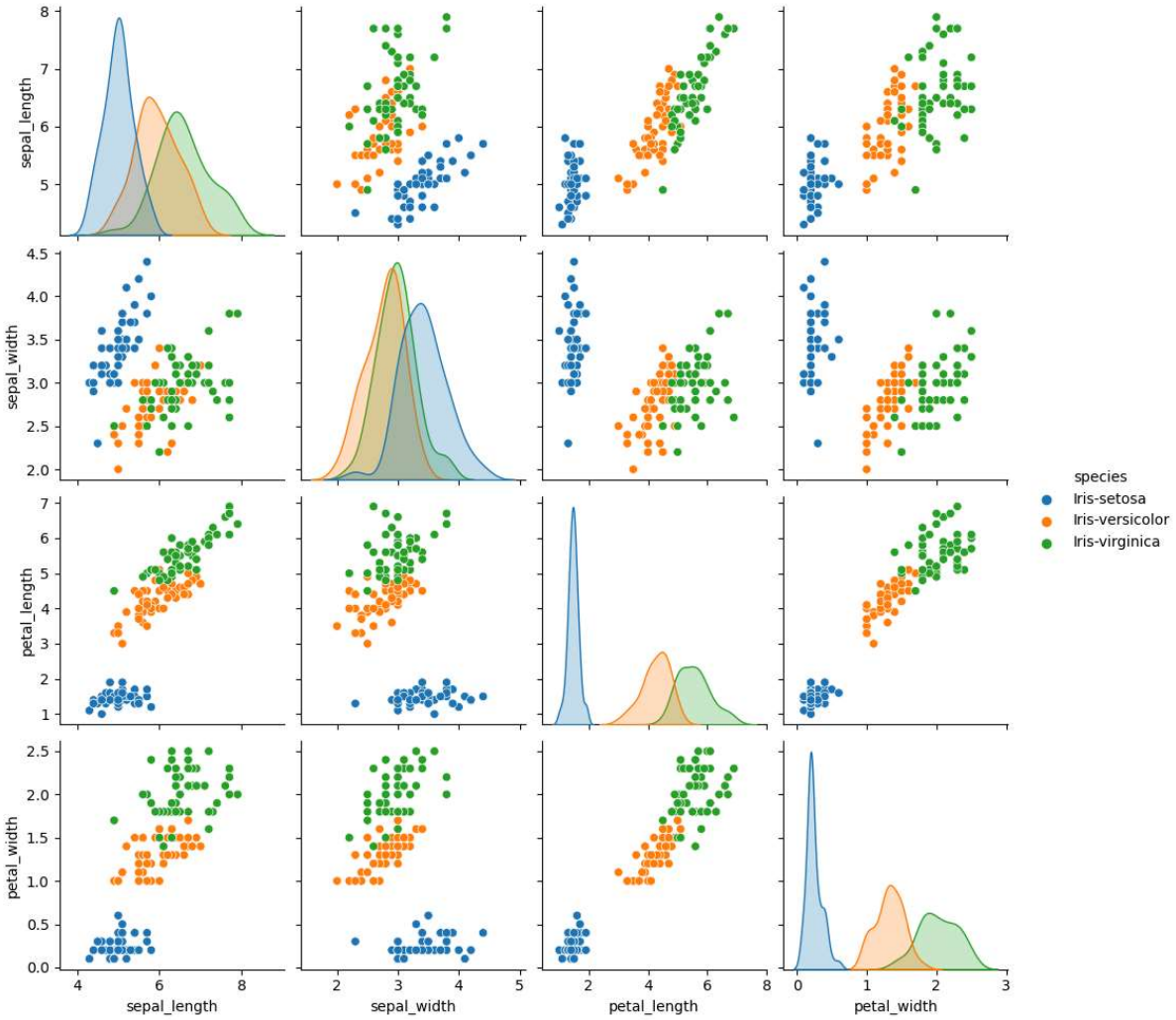
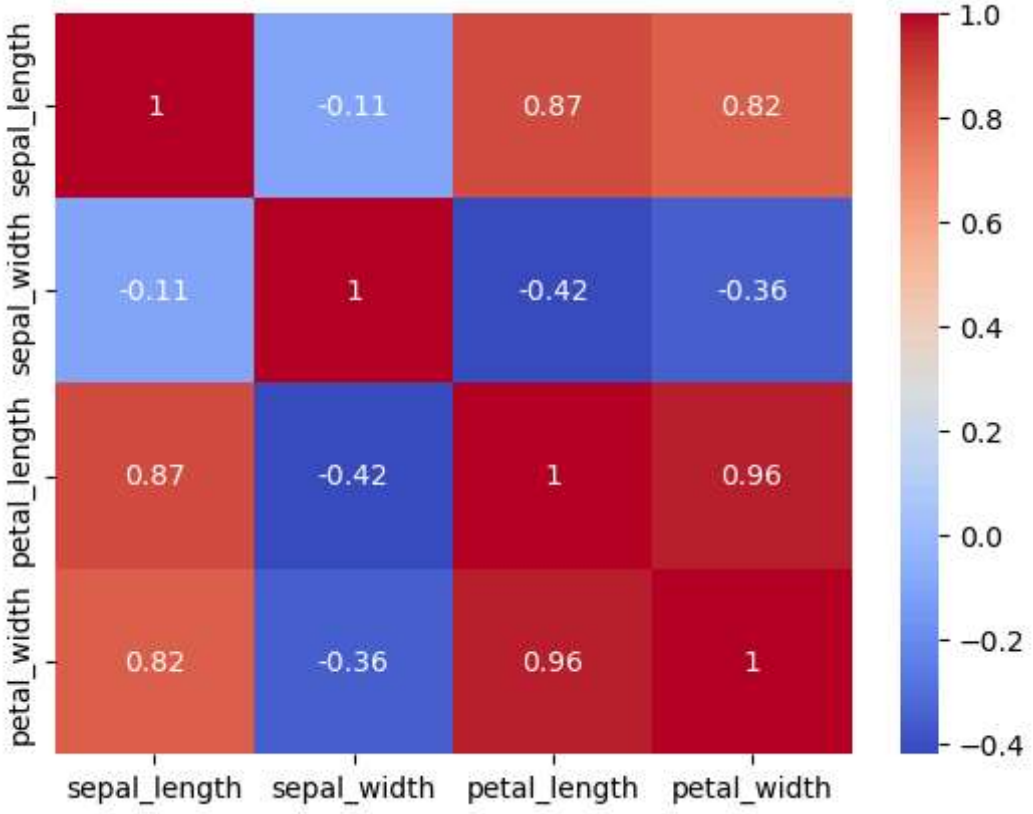
In [2]:

```
# Created by Lettie Ngobeni

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score


data = pd.read_csv('IRIS.csv')

print("Displaying the initial rows of the dataset:")
print(data.head())

# ANALYSE
print("\nChecking for missing values in the dataset:")
print(data.isnull().sum())

print("\nGenerating Data Visualizations:")


sns.pairplot(data, hue='species')
plt.show()

numeric_data = data.drop('species', axis=1)
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.show()

#  Data Preprocessing

le = LabelEncoder()
data['species'] = le.fit_transform(data['species'])

print("\nDataset after encoding the 'species' column:")
print(data.head())

# Scaling the feature variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(data.drop('species', axis=1))
X_scaled = pd.DataFrame(X_scaled, columns=data.columns[:-1])
print("\nScaled features:")
print(X_scaled.head())


X_scaled['species'] = data['species']

# Model Development
X = X_scaled.drop('species', axis=1)
y = X_scaled['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)


#  test set
y_pred = model.predict(X_test)

# Evaluating model performance
print("\nEvaluation of the RandomForest Model:")
```

```
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nModel Accuracy:")
print(accuracy_score(y_test, y_pred))

#  predictions on new data

new_observation = [[5.1, 3.5, 1.4, 0.2]]  # Example input
scaled_observation = scaler.transform(new_observation)
scaled_observation_df = pd.DataFrame(scaled_observation, columns=X.columns)
predicted_species = model.predict(scaled_observation_df)
final_prediction = le.inverse_transform(predicted_species)
print(f'\nPredicted species for the new observation: {final_prediction[0]}')
```

```
Displaying the initial rows of the dataset:
   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

Checking for missing values in the dataset:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64

Generating Data Visualizations:
C:\Users\nhlav\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: T
he figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```


sepal_length
sepal_width
petal_length
petal_width
species
Iris-setosa
Iris-versicolor
Iris-virginica


sepal_length
sepal_width
petal_length
petal_width
1
-0.11
0.87
0.82
-0.11
1
-0.42
-0.36
0.87
-0.42
1
0.96
0.82
-0.36
0.96
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

```
Dataset after encoding the 'species' column:
   sepal_length  sepal_width  petal_length  petal_width  species
0           5.1          3.5           1.4          0.2        0
1           4.9          3.0           1.4          0.2        0
2           4.7          3.2           1.3          0.2        0
3           4.6          3.1           1.5          0.2        0
4           5.0          3.6           1.4          0.2        0

Scaled features:
   sepal_length  sepal_width  petal_length  petal_width
0     -0.900681     1.032057     -1.341272    -1.312977
1     -1.143017    -0.124958     -1.341272    -1.312977
2     -1.385353     0.337848     -1.398138    -1.312977
3     -1.506521     0.106445     -1.284407    -1.312977
4     -1.021849     1.263460     -1.341272    -1.312977

Evaluation of the RandomForest Model:

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


Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Model Accuracy:
1.0

Predicted species for the new observation: Iris-setosa
C:\Users\nhlav\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X doe
s not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

In [ ]: