In [29]:
```
pip install numpy pandas scikit-learn matplotlib seaborn
```

```
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: pandas in c:\users\lenovo\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: scikit-learn in c:\users\lenovo\anaconda3\lib\site-packages (1.5.1)
Requirement already satisfied: matplotlib in c:\users\lenovo\anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\lenovo\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\lenovo\anaconda3\lib\site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\lenovo\anaconda3\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\lenovo\anaconda3\lib\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

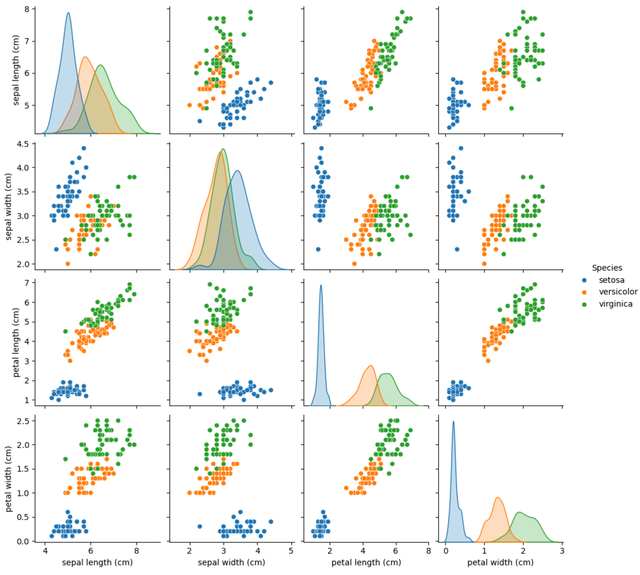
In [3]:
```
from sklearn.datasets import load_iris
import pandas as pd

# Load dataset
data = load_iris()

# Create a pandas DataFrame
df = pd.DataFrame(data.data, columns=data.feature_names)

# Add the target labels (species)
df['Species'] = pd.Categorical.from_codes(data.target, data.target_names)

# Display the first few rows of the dataset
print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  Species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa
```

In [5]:
```
print(df.describe())
```

```
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000
mean            5.843333          3.057333           3.758000
std             0.828066          0.435866           1.765298
min             4.300000          2.000000           1.000000
25%             5.100000          2.800000           1.600000
50%             5.800000          3.000000           4.350000
75%             6.400000          3.300000           5.100000
max             7.900000          4.400000           6.900000

       petal width (cm)
count        150.000000
mean           1.199333
std            0.762238
min            0.100000
25%            0.300000
50%            1.300000
75%            1.800000
max            2.500000
```

In [7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Pairplot for visualizing relationships between features
sns.pairplot(df, hue='Species')
plt.show()
```



In [9]:
```
X = df.drop('Species', axis=1)
y = df['Species']
```

In [11]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [13]:
```
from sklearn.neighbors import KNeighborsClassifier

# Initialize the model with 3 neighbors
knn = KNeighborsClassifier(n_neighbors=3)

# Train the model
knn.fit(X_train, y_train)
```

Out[13]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [15]:
```
from sklearn.metrics import accuracy_score

# Predict the species of the test set
y_pred = knn.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
Accuracy: 100.00%
```

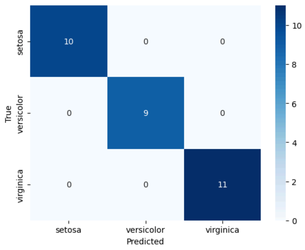
In [17]:
```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)
```

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

In [19]:
```
from sklearn.model_selection import GridSearchCV

# Define the parameter grid
param_grid = {'n_neighbors': [1, 3, 5, 7, 9]}

# Grid search for the best number of neighbors
grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5)
grid_search.fit(X_train, y_train)

print(f"Best number of neighbors: {grid_search.best_params_}")
```

```
Best number of neighbors: {'n_neighbors': 3}
```

In [21]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Heatmap for Confusion Matrix
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=data.target_names, yticklabels=data.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```



In [23]:
```
def predict_species(sepal_length, sepal_width, petal_length, petal_width):
    new_data = [[sepal_length, sepal_width, petal_length, petal_width]]
    prediction = knn.predict(new_data)
    return prediction[0]

# Example prediction
print(predict_species(5.1, 3.5, 1.4, 0.2))
```

```
setosa
C:\Users\LENOVO\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
  warnings.warn(
```

In [27]:
```
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 (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   Species            150 non-null    category
```

```
dtypes: category(1), float64(4)
memory usage: 5.1 KB
```

In [ ]: