In [1]:
```
# 1. Import Required Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.decomposition import PCA
```

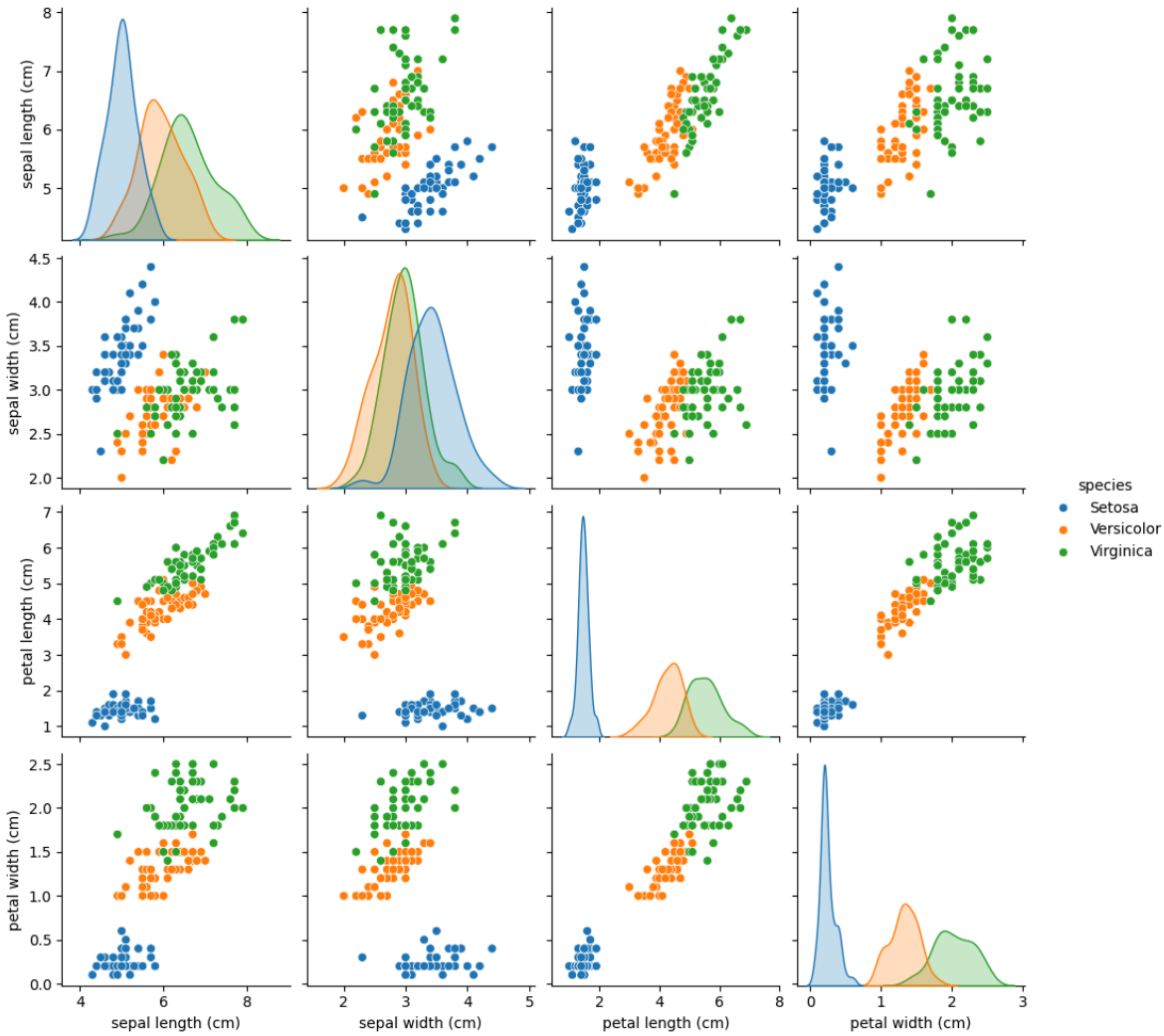
2. Dataset Loading: Load the Iris dataset and understand its structure

In [2]:
```
iris = load_iris()
data = pd.DataFrame(iris.data, columns=iris.feature_names)
data['species'] = iris.target
data['species'] = data['species'].map({0:'Setosa',1:'Versicolor', 2:'Virginica
print(data.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 [3]:
```
sns.pairplot(data, hue='species',diag_kind='kde')
plt.show()
```



Data Preprocessing

In [9]:
```
X= data.iloc[:, :-1]
y = data['species']
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
```

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.3
```

Model Training

In [11]:
```
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
```

Out[11]:
```
RandomForestClassifier(random_state=42)
```
(https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble

Model Evaluation

In [12]:
```
y_pred = rf_model.predict(X_test)

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Confusion Matrix:
[[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]

Classification Report:
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        19
  Versicolor       1.00      1.00      1.00        13
   Virginica       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

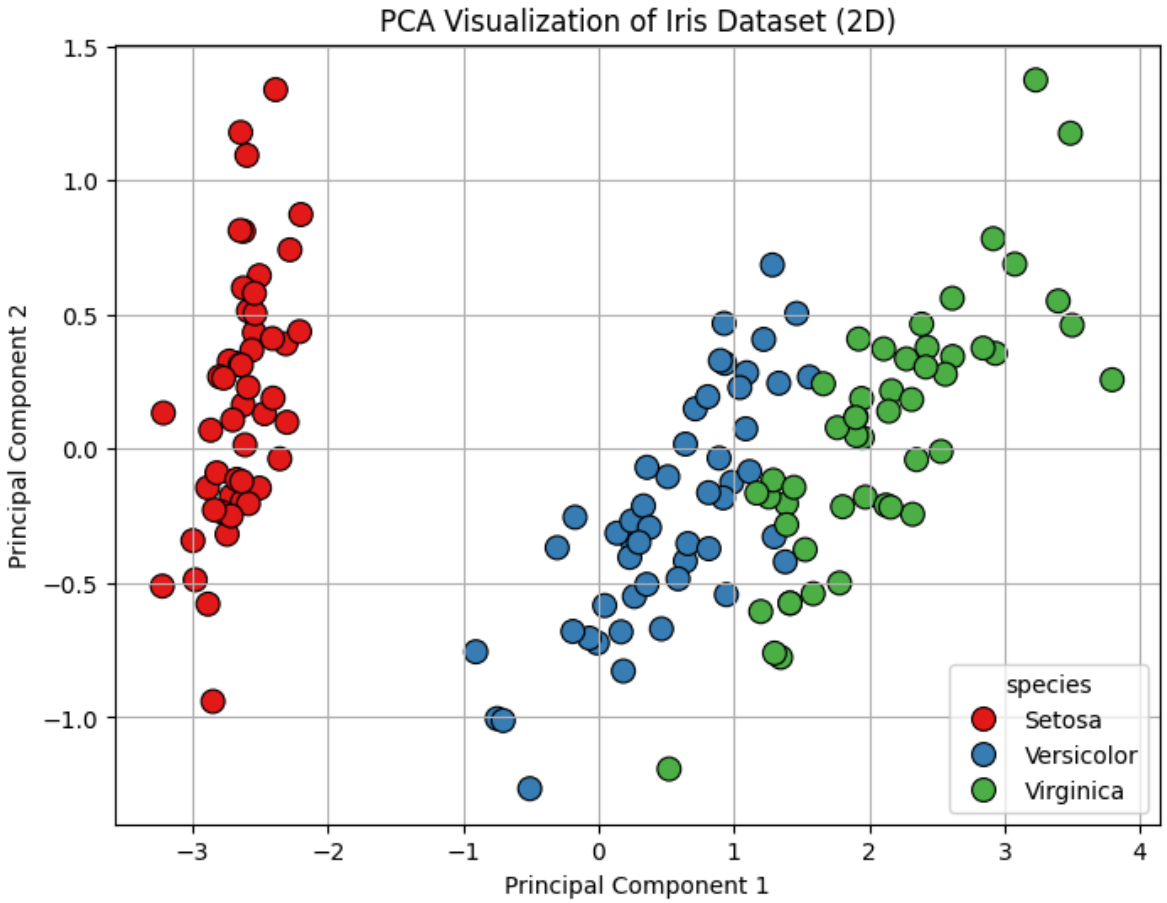
Visualization: PCA to reduce dimensions and plot unique species patterns

In [13]:
```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

pca_df = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
pca_df['species'] = y.values
```

In [14]:

```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=pca_df, x='PC1', y='PC2', hue='species', palette='Set1',
plt.title('PCA Visualization of Iris Dataset (2D)')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.grid(True)
plt.show()
```



In [ ]: