In [1]:

```
!pip install numpy pandas scikit-learn matplotlib seaborn
```

```
Requirement already satisfied: numpy in c:\users\shouvik\anaconda3\lib\site-packages
(2.1.3)
Requirement already satisfied: pandas in c:\users\shouvik\anaconda3\lib\site-package
s (2.2.3)
Requirement already satisfied: scikit-learn in c:\users\shouvik\anaconda3\lib\site-p
ackages (1.6.1)
Requirement already satisfied: matplotlib in c:\users\shouvik\anaconda3\lib\site-pac
kages (3.10.0)
Requirement already satisfied: seaborn in c:\users\shouvik\anaconda3\lib\site-packag
es (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\shouvik\anaconda3
\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\shouvik\anaconda3\lib\site-p
ackages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\shouvik\anaconda3\lib\site
-packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\shouvik\anaconda3\lib\site-p
ackages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in c:\users\shouvik\anaconda3\lib\site-
packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\shouvik\anaconda3\li
b\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\shouvik\anaconda3\lib\si
te-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\shouvik\anaconda3\lib\site-p
ackages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\shouvik\anaconda3\lib\s
ite-packages (from matplotlib) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\shouvik\anaconda3\lib\s
ite-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\shouvik\anaconda3\lib\sit
e-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\shouvik\anaconda3\lib\site-pack
ages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\shouvik\anaconda3\lib\si
te-packages (from matplotlib) (3.2.0)
Requirement already satisfied: six>=1.5 in c:\users\shouvik\anaconda3\lib\site-packa
ges (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

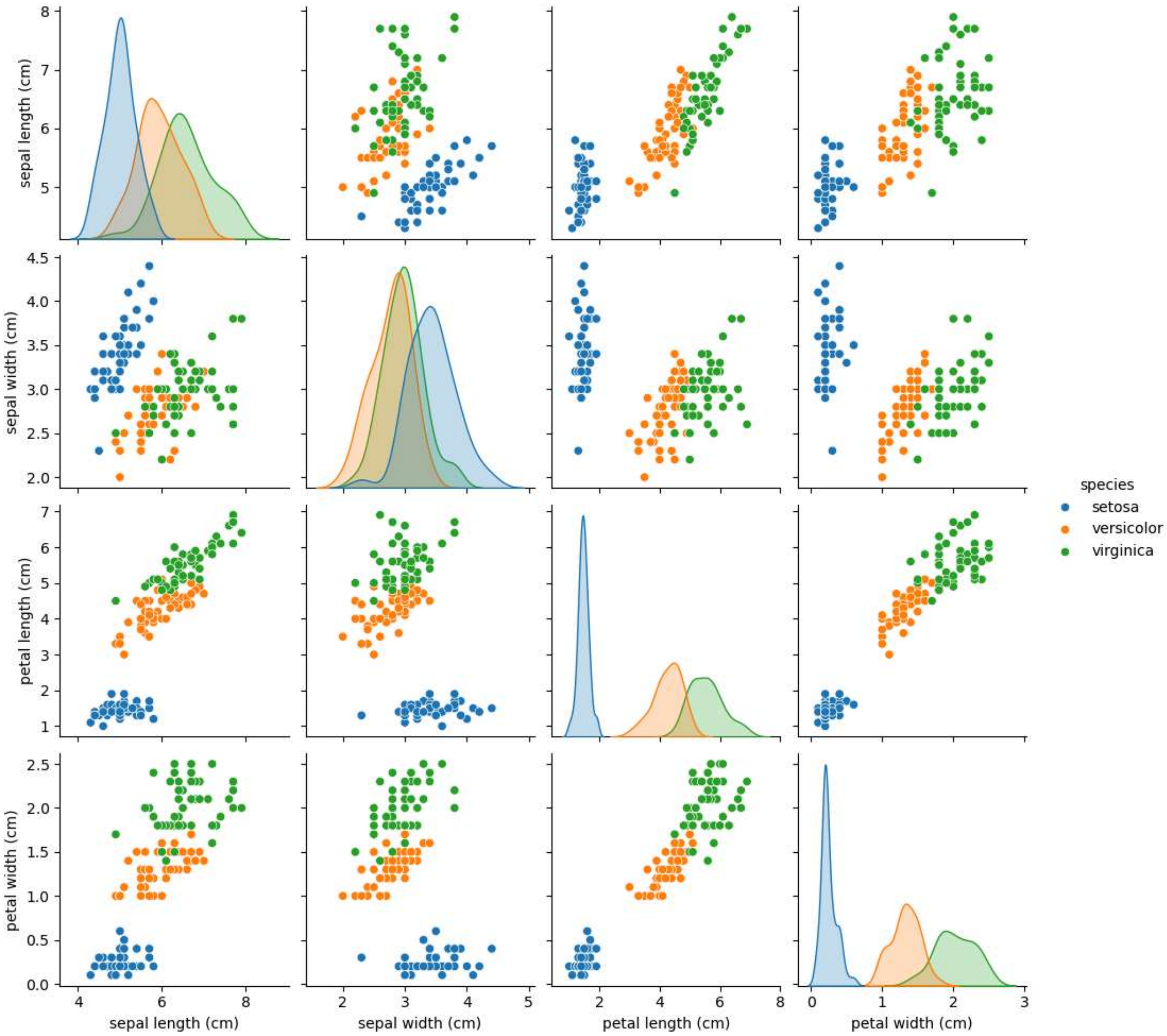
In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDispla
```

In [3]:

```
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target, name='species')

df = X.copy()
df['species'] = y.map({i: name for i, name in enumerate(iris.target_names)})

print(df.head())
print(df['species'].value_counts())

# Visualization
sns.pairplot(df, hue='species')
plt.show()
```

```
   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  
species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```



In [4]:
```python
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [5]:
```python
log_model = LogisticRegression()
log_model.fit(X_train_scaled, y_train)

y_pred_log = log_model.predict(X_test_scaled)
acc_log = accuracy_score(y_test, y_pred_log)
print(f"Logistic Regression Accuracy: {acc_log:.2f}")
```

```
Logistic Regression Accuracy: 1.00
```

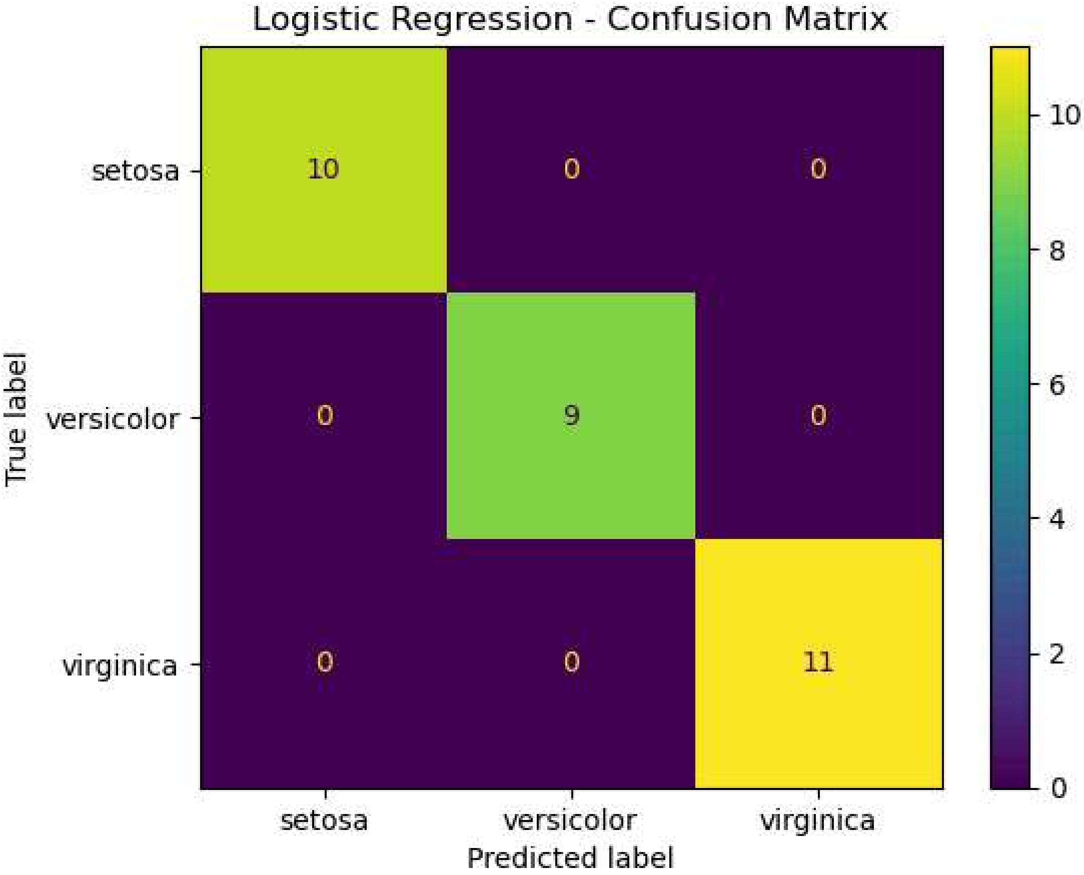
In [6]:
```python
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train_scaled, y_train)

y_pred_knn = knn_model.predict(X_test_scaled)
```

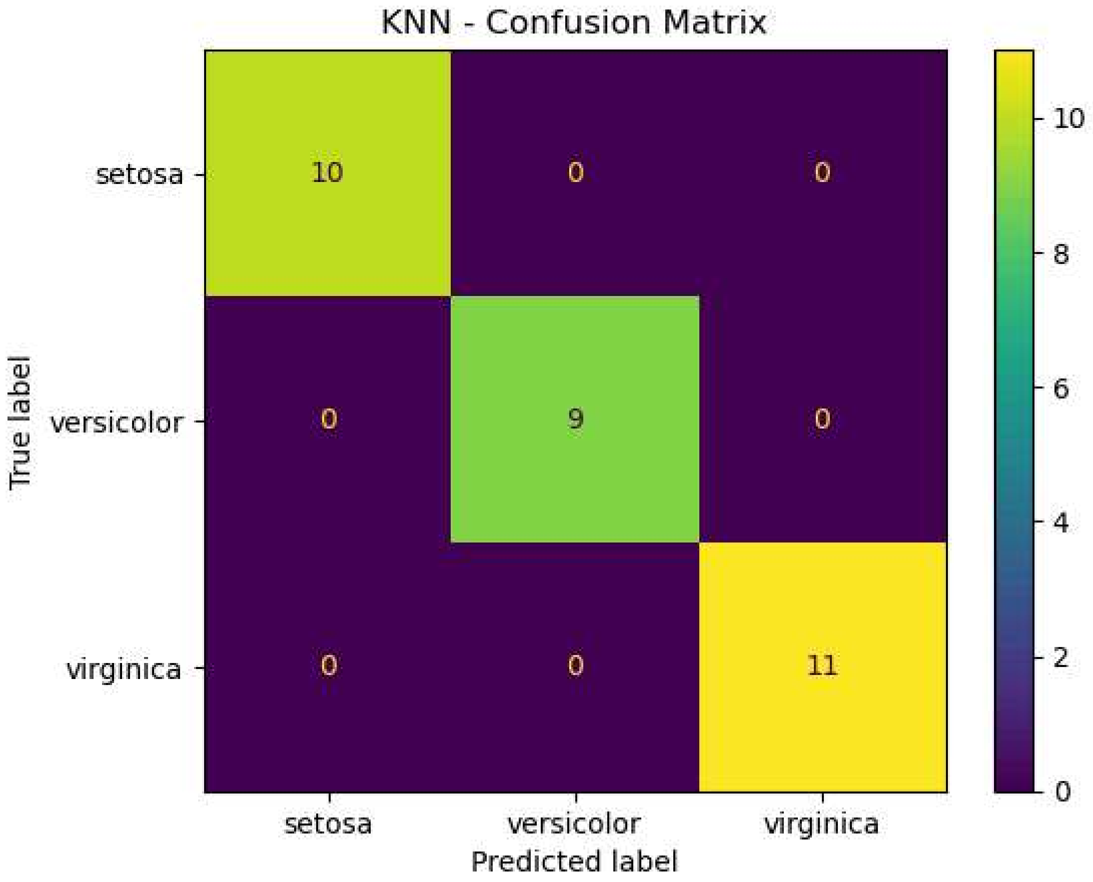
```
acc_knn = accuracy_score(y_test, y_pred_knn)
print(f"KNN Accuracy: {acc_knn:.2f}")
```

```
KNN Accuracy: 1.00
```

In [7]:

```
# Logistic Regression
cm_log = confusion_matrix(y_test, y_pred_log)
ConfusionMatrixDisplay(cm_log, display_labels=iris.target_names).plot()
plt.title("Logistic Regression - Confusion Matrix")
plt.show()

# KNN
cm_knn = confusion_matrix(y_test, y_pred_knn)
ConfusionMatrixDisplay(cm_knn, display_labels=iris.target_names).plot()
plt.title("KNN - Confusion Matrix")
plt.show()
```





📝 Task 1 Summary Report – Iris Classification

Objective:

To understand basic machine learning classification using the Iris dataset, applying logistic regression and K-nearest neighbors (KNN) algorithms, and evaluating performance using accuracy and confusion matrix.

Dataset Used:

- Iris dataset from `sklearn.datasets`
- 150 samples, 4 features: sepal length, sepal width, petal length, petal width
- 3 classes: `setosa` , `versicolor` , `virginica`

Steps Followed:

1. Exploratory Data Analysis (EDA) using pandas and seaborn
2. Data Preprocessing: Train-test split, StandardScaler
3. Model Training:

   - Logistic Regression
   - K-Nearest Neighbors (KNN) with k=3
4. Evaluation:
   - Accuracy Score
   - Confusion Matrix

Results:

Model	Accuracy
Logistic Regression	1.00
KNN (k=3)	1.00

Conclusion:

- Both logistic regression and KNN performed very well on the Iris dataset.
- KNN was slightly easier to implement and gave the same result.
- The simplicity of the dataset makes it ideal for beginner ML tasks.

```
In [ ]:
```