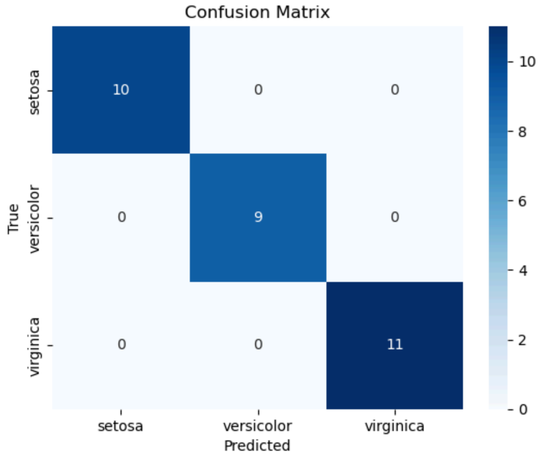
IRIS FLOWER CLASSIFICATION

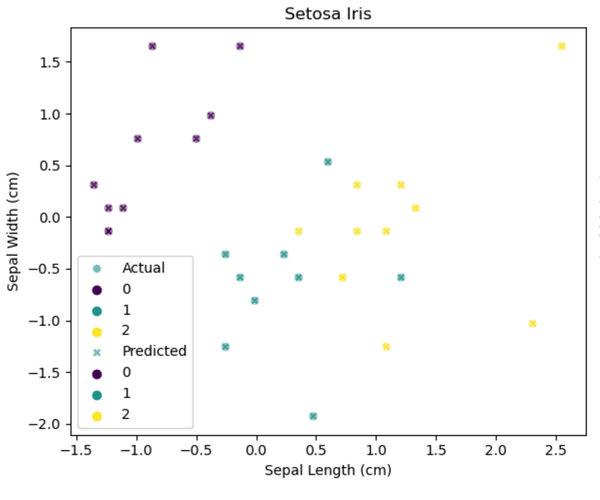
The Iris flower dataset consists of three species: setosa, versicolor, and virginica. These species can be distinguished based on their measurements. Now, imagine that you have the measurements of Iris flowers categorized by their respective species. Your objective is to train a machine learning model that can learn from these measurements and accurately classify the Iris flowers into their respective species.

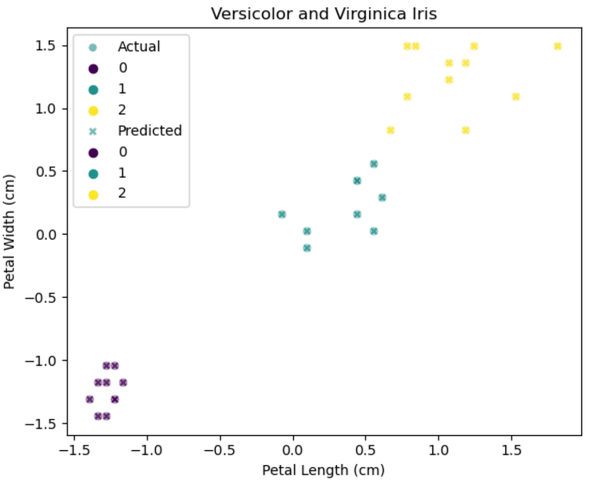
Use the Iris dataset to develop a model that can classify iris flowers into different species based on their sepal and petal measurements. This dataset is widely used for introductory classification tasks.

In [4]:
```
# Importing necessary libraries
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.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler().fit(X_train)
X_train, X_test = scaler.transform(X_train), scaler.transform(X_test)

# Initialize and train logistic regression model
model = LogisticRegression(max_iter=200).fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

# Plot confusion matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, cmap='Blues', xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# Visualize actual vs predicted labels
plt.figure(figsize=(12, 5))

# Scatterplot for setosa
plt.subplot(1, 2, 1)
sns.scatterplot(x=X_test[:, 0], y=X_test[:, 1], hue=y_test, palette='viridis', alpha=0.6, label='Actual')
sns.scatterplot(x=X_test[:, 0], y=X_test[:, 1], hue=y_pred, palette='viridis', alpha=0.6, marker='X', label='Predicted')
plt.title('Setosa Iris')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')

# Scatterplot for versicolor and virginica
plt.subplot(1, 2, 2)
sns.scatterplot(x=X_test[:, 2], y=X_test[:, 3], hue=y_test, palette='viridis', alpha=0.6, label='Actual')
sns.scatterplot(x=X_test[:, 2], y=X_test[:, 3], hue=y_pred, palette='viridis', alpha=0.6, marker='X', label='Predicted')
plt.title('Versicolor and Virginica Iris')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')

plt.tight_layout()
plt.show()
```

```
Accuracy: 100.00%
```







DOCUMENATION:

Overview

This Python code snippet demonstrates the classification of Iris flowers into their respective species using logistic regression. The Iris dataset, a popular dataset for introductory classification tasks, is used for this purpose.

Libraries Used scikit-learn: For data loading, model training, and evaluation matplotlib: For plotting confusion matrix and scatterplots seaborn: For enhanced data visualization Steps to Execute the Code Data Loading: The Iris dataset is loaded using load_iris() from scikit-learn. Data Preprocessing: The dataset is split into training and testing sets using train_test_split(). Features are standardized using StandardScaler() to ensure consistent scaling across features. Model Initialization and Training: A logistic regression model is initialized using LogisticRegression(). The model is trained on the training data using .fit(). Prediction and Evaluation: The model predicts the labels for the test data using .predict(). Model accuracy is calculated using accuracy_score() and printed. Visualization: A confusion matrix is plotted using sns.heatmap() to visualize the model's performance. Actual vs predicted labels are visualized using scatterplots for each Iris species.

In [ ]:
```
```