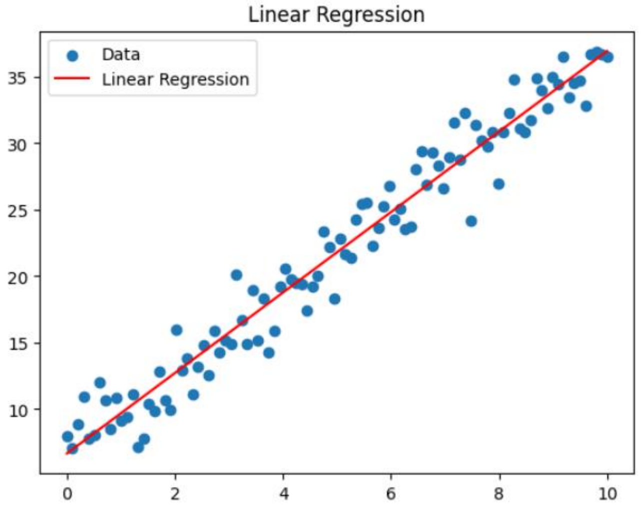

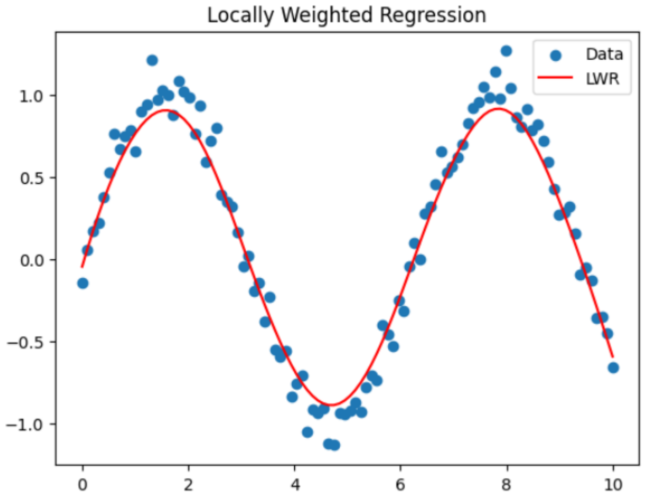

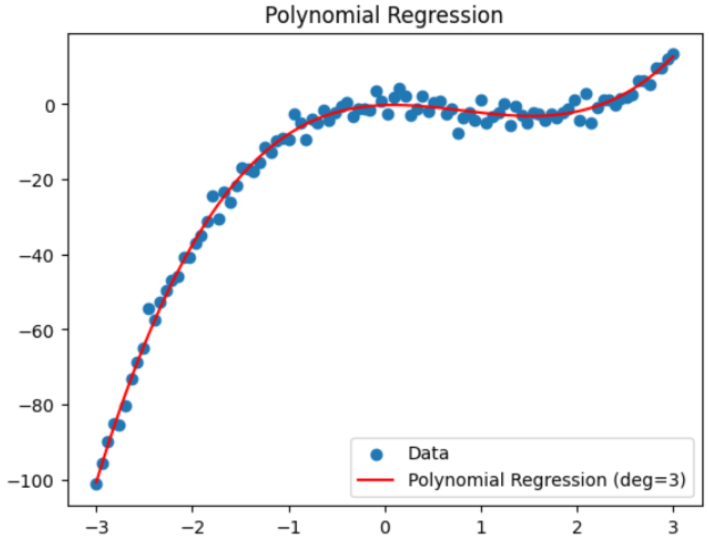

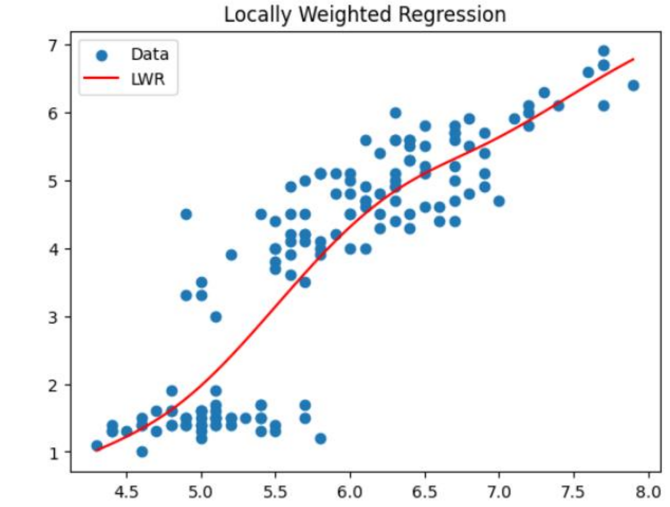

[19]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from scipy.spatial.distance import cdist
df_lwr =pd.read_csv("C:/Users/Prakrithi/Downloads/iris - iris.csv")
# X, y = df[['sepal.length']], df['petal.length']
X = df_lwr[['sepal.length']].values
y = df_lwr['petal.length'].values
def gaussian_kernel(x, X, tau):
    return np.exp(-cdist([[x]], X, 'sqeuclidean') / (2 * tau**2))
def locally_weighted_regression(X_train, y_train, tau=0.5):
    X_train = np.hstack([np.ones((X_train.shape[0], 1)), X_train])
    X_range = np.linspace(X_train[:, 1].min(), X_train[:, 1].max(), 100)
    y_pred = []
    for x in X_range:
        x_vec = np.array([1, x])  # Intercept term
        weights = gaussian_kernel(x, X_train[:, 1:], tau).flatten()
        W = np.diag(weights)
        theta = np.linalg.pinv(X_train.T @ W @ X_train) @ (X_train.T @ W @ y_train)
        y_pred.append(x_vec @ theta)
    plt.scatter(X_train[:, 1], y_train, label='Data')
    plt.plot(X_range, y_pred, color='red', label='LWR')
    plt.legend()
    plt.title("Locally Weighted Regression")
    plt.show()
locally_weighted_regression(df_lwr[['sepal.length']].values, df_lwr['petal.length'].values)
```