Step 1: Import necessary libraries and set up random data generation.

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 1 + 2 * X + np.random.randn(100, 1)
```

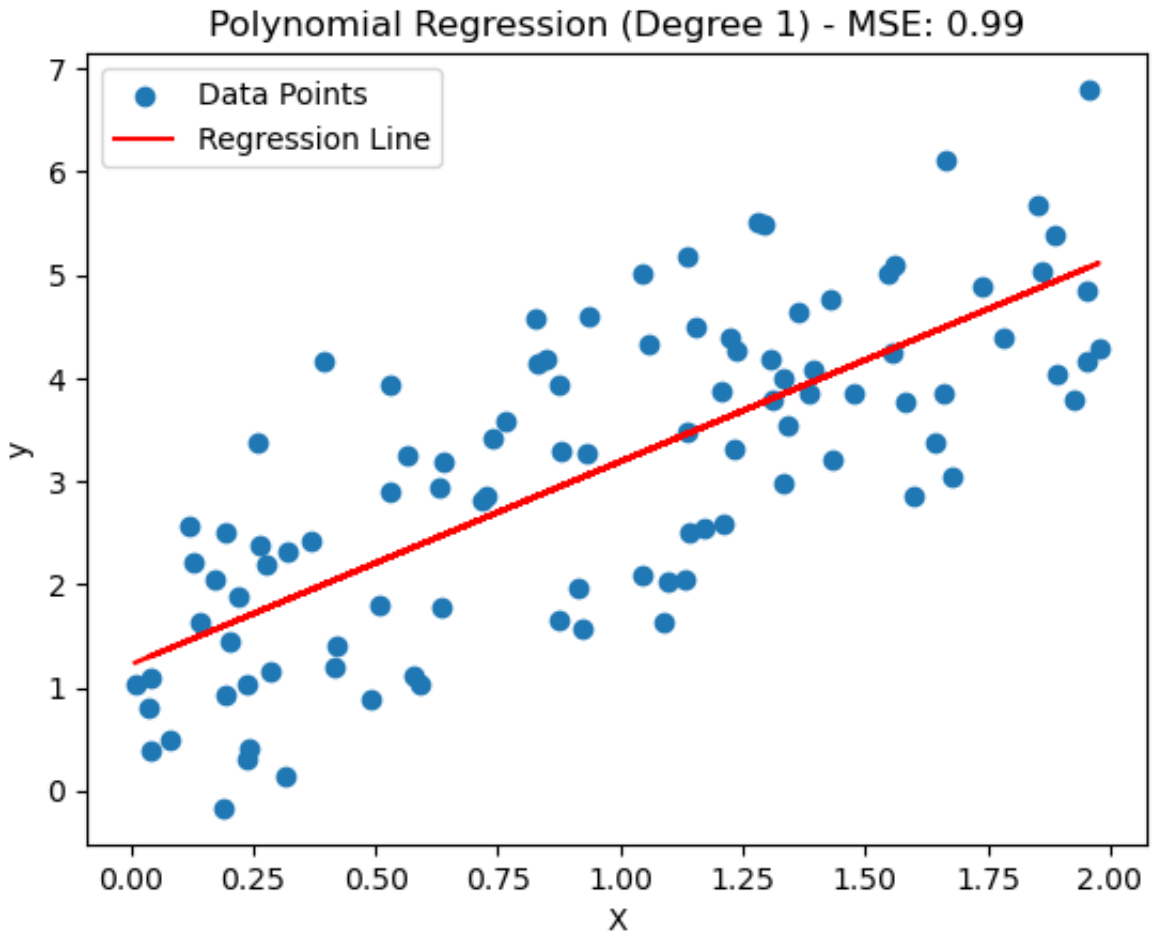
Step 2: Define a function to plot the data and regression lines.

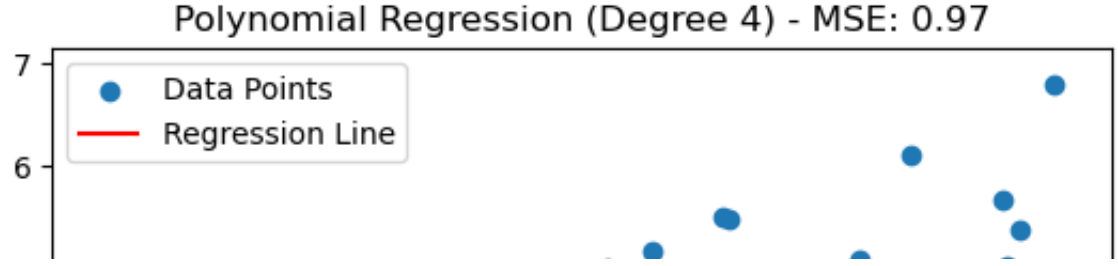
In [3]:
```
def plot_data_and_model(X, y, y_pred, title):
    plt.scatter(X, y, label='Data Points')
    plt.plot(X, y_pred, color='red', label='Regression Line')
    plt.title(title)
    plt.xlabel('X')
    plt.ylabel('y')
    plt.legend()
    plt.show()
```

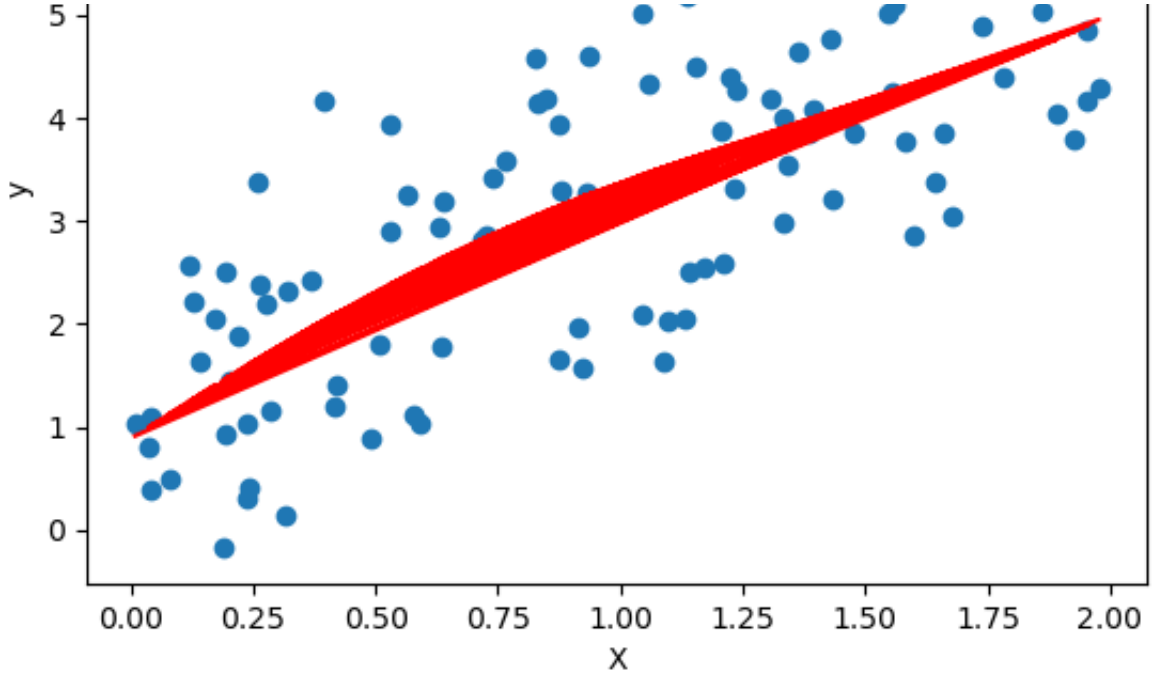
Step 3: Create polynomial regression models with different degrees (complexity) and observe the fit.

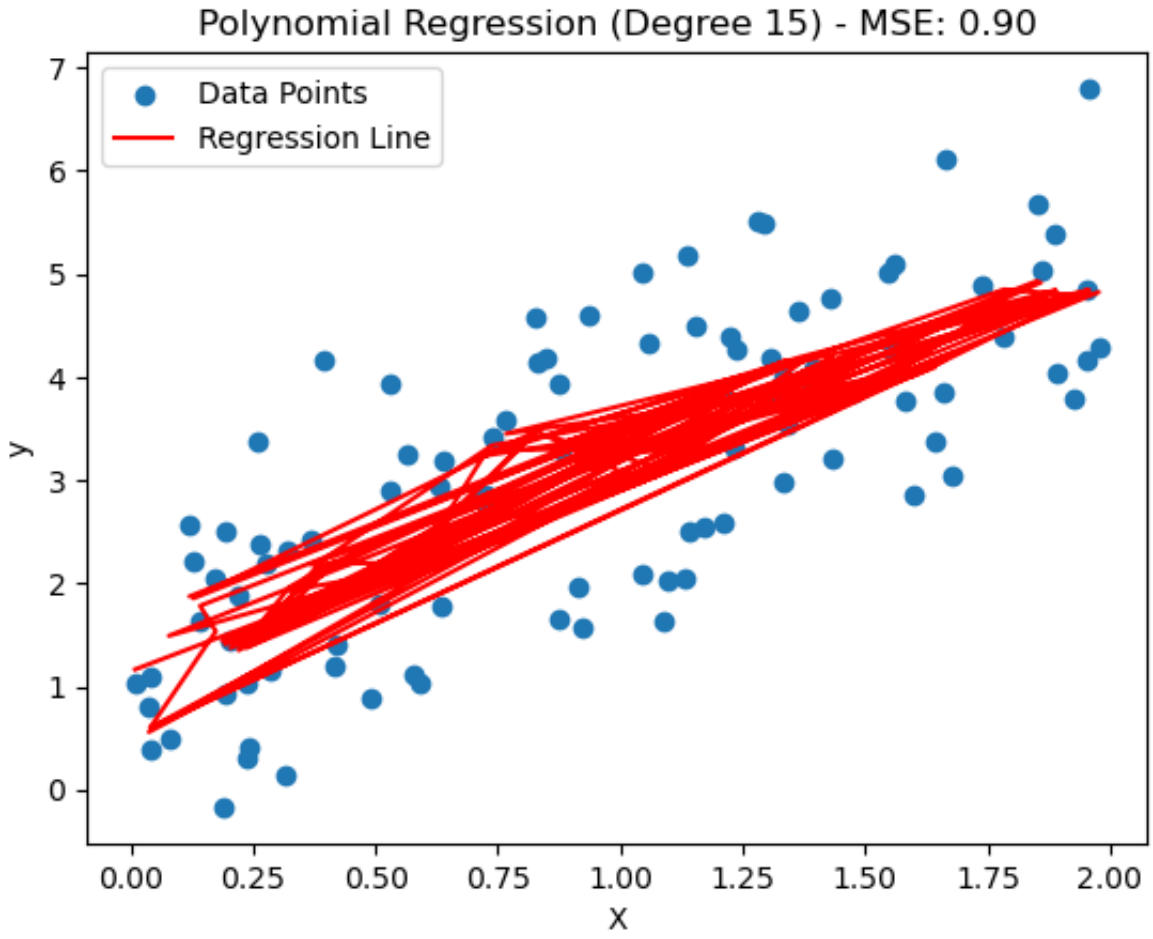
In [4]:
```

```

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error

degrees = [1, 4, 15]  # Different degrees of polynomial regression

for degree in degrees:
    polynomial_features = PolynomialFeatures(degree=degree)
    X_poly = polynomial_features.fit_transform(X)

    model = LinearRegression()
    model.fit(X_poly, y)

    y_pred = model.predict(X_poly)

    mse = mean_squared_error(y, y_pred)

    plot_data_and_model(X, y, y_pred, f'Polynomial Regression (Degr
```






y
X
0
1
2
3
4
5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Polynomial Regression (Degree 15) - MSE: 0.90
Data Points
Regression Line
y
X
0
1
2
3
4
5
6
7
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00