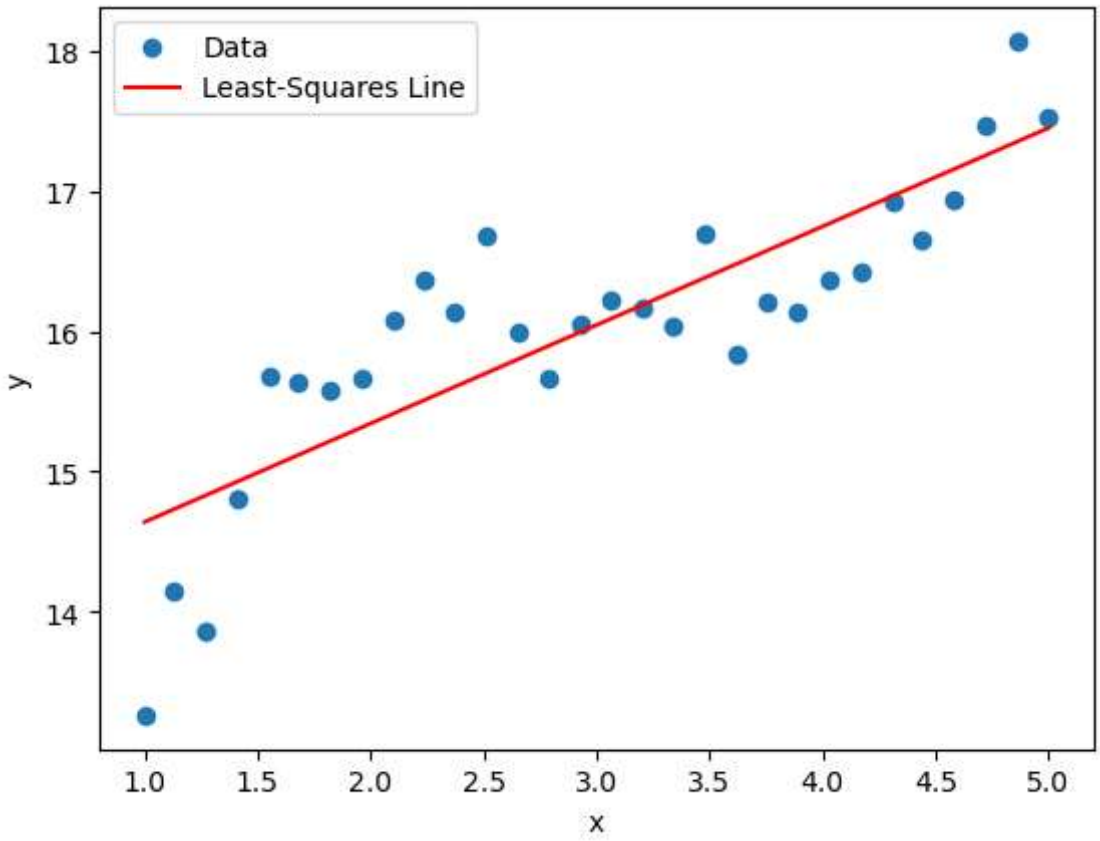
1

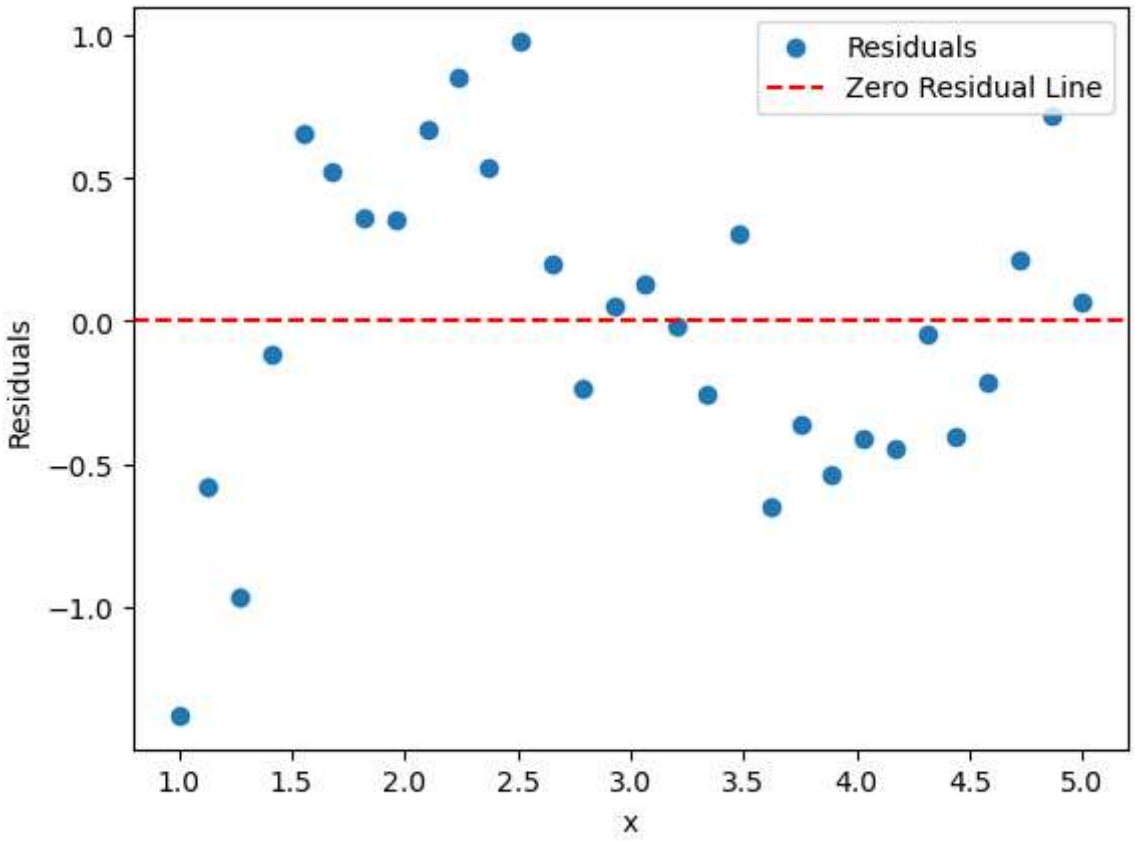
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Given data
y = np.array([13.26, 14.15, 13.86, 14.81, 15.68, 15.64, 15.58, 15.67, 16.08, 16.36, 16
              16.68, 16.00, 15.66, 16.05, 16.22, 16.17, 16.03, 16.69, 15.83, 16.21, 16
              16.36, 16.42, 16.92, 16.65, 16.94, 17.47, 18.07, 17.52])

x = np.array([1.00, 1.13, 1.27, 1.41, 1.55, 1.68, 1.82, 1.96, 2.10, 2.24, 2.37, 2.51,
              2.79, 2.93, 3.06, 3.20, 3.34, 3.48, 3.62, 3.75, 3.89, 4.03, 4.17, 4.31,
              4.58, 4.72, 4.86, 5.00])

# Calculating the least-squares line coefficient
n = len(x)
sum_x = np.sum(x)
sum_y = np.sum(y)
sum_xy = np.sum(x * y)
sum_x_squared = np.sum(x ** 2)

# Slope (m) and intercept (b) formulas
m = (n * sum_xy - sum_x * sum_y) / (n * sum_x_squared - sum_x ** 2)
b = (sum_y - m * sum_x) / n

# Least-squares line equation
line_eq = lambda x: m * x + b

# Plotting the data and the least-squares line
plt.scatter(x, y, label='Data')
plt.plot(x, line_eq(x), color='red', label='Least-Squares Line')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()

print("Slope (m):", m)
print("Intercept (b):", b)
```




```
Slope (m): 0.7029398624005561
Intercept (b): 13.936127465489532
```

2

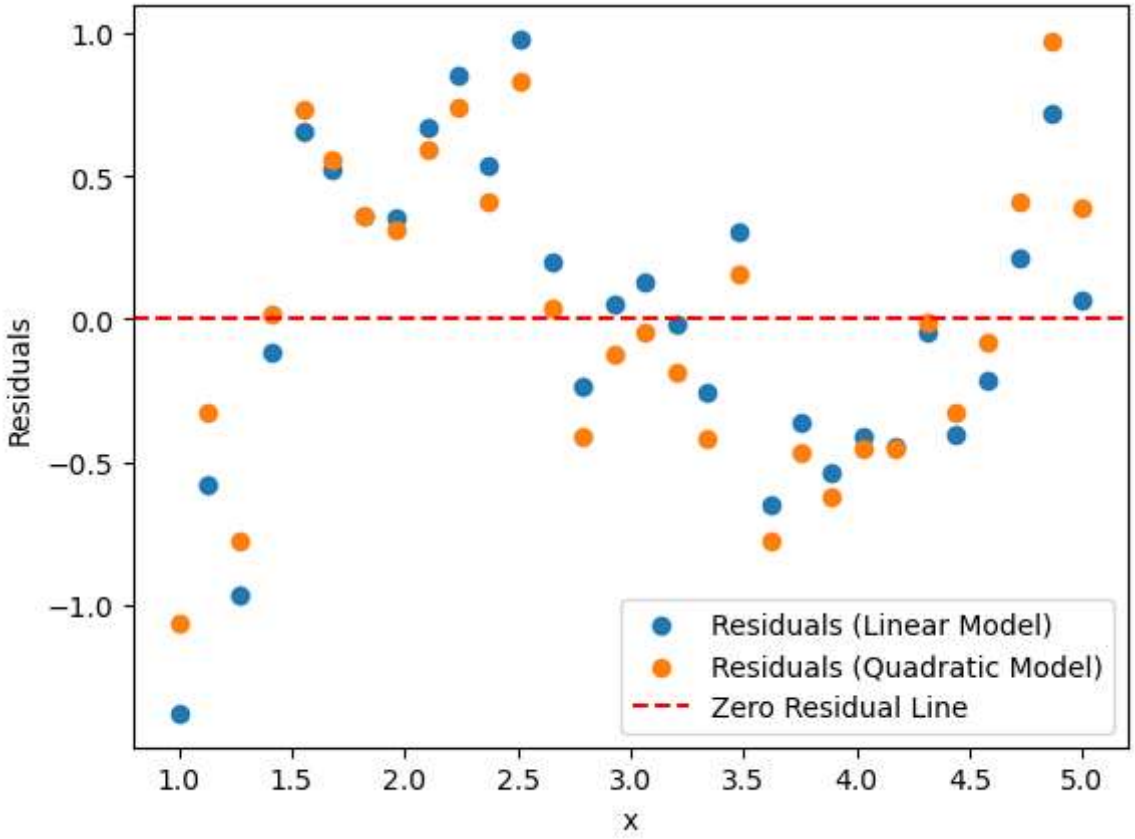
In [4]:
```
# Calculate residuals
residuals = y - (m * x + b)

# Plot residuals
plt.scatter(x, residuals, label='Residuals')
plt.axhline(y=0, color='red', linestyle='--', label='Zero Residual Line')
plt.xlabel('x')
plt.ylabel('Residuals')
plt.legend()
plt.show()
```



If the residuals exhibit a random pattern around zero, it indicates that the model is a good fit. If there is a pattern or trend, it suggests that the model may not be suitable.

3

In [5]:
```
# Fit a quadratic model
coefficients = np.polyfit(x, y, 2)
quadratic_fit = np.poly1d(coefficients)

# Calculate residuals for the quadratic model
residuals_quadratic = y - quadratic_fit(x)

# Plot residuals for both models
plt.scatter(x, residuals, label='Residuals (Linear Model)')
plt.scatter(x, residuals_quadratic, label='Residuals (Quadratic Model)')
plt.axhline(y=0, color='red', linestyle='--', label='Zero Residual Line')
plt.xlabel('x')
plt.ylabel('Residuals')
plt.legend()
plt.show()
```



residuals for the linear model and the quadratic model to assess which one provides a better fit

In [ ]: