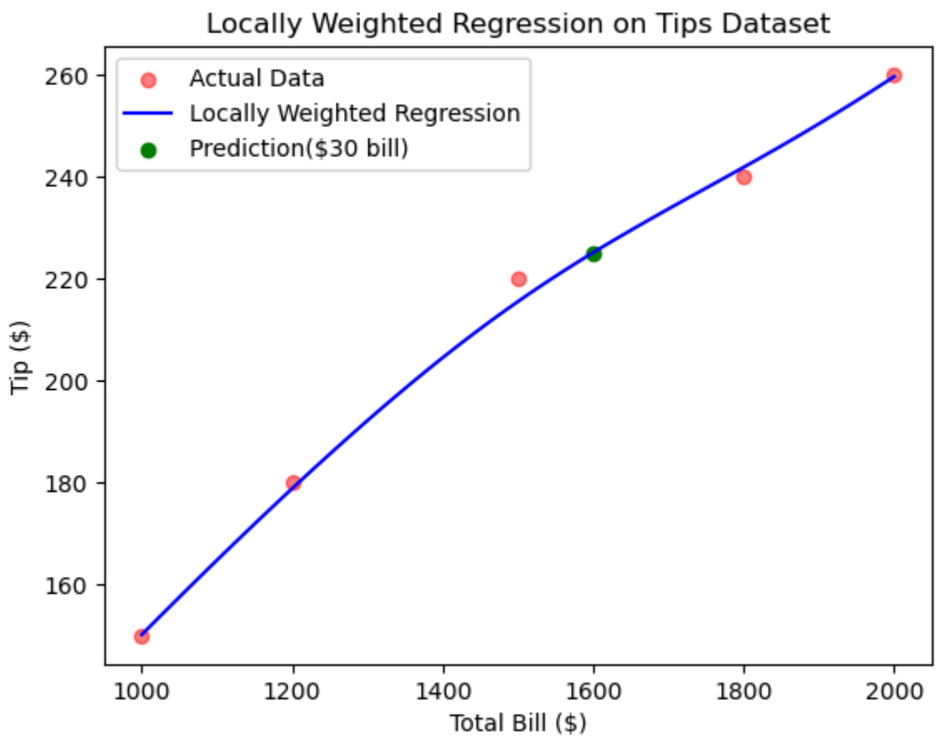
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
file_path = "house.csv" # Update this with the actual path to your CSV file
tips = pd.read_csv(file_path)
```

In [4]:
```
X = tips["House Size"].values
y = tips["Price"].values
```

In [6]:
```
def locally_weighted_regression(x_query, X, y, tau):
    m = len(X)
    weights = np.exp(-((X - x_query) ** 2) / (2 * tau ** 2))
    X_b = np.c_[np.ones(m), X] # Add bias term (column of ones)
    W = np.diag(weights) # Create diagonal weight matrix
    theta = np.linalg.pinv(X_b.T @ W @ X_b) @ (X_b.T @ W @ y)
    x_query_b = np.array([1, x_query])
    return x_query_b @ theta
tau = 200 # Bandwidth (adjust as needed)
x_query =1600
predicted_tip = locally_weighted_regression(x_query, X, y, tau)
print(f"Predicted Tip for a total bill of $30: {predicted_tip:.2f}")
X_range = np.linspace(X.min(), X.max(), 100)
y_pred = np.array([locally_weighted_regression(x, X, y, tau) for x in X_range])
plt.scatter(X, y, color='red', alpha=0.5, label="Actual Data")
plt.plot(X_range, y_pred, color='blue', label="Locally Weighted Regression")
plt.scatter([x_query], [predicted_tip], color='green', marker='o', label="Prediction($30 bill)")
plt.xlabel("Total Bill ($)")
plt.ylabel("Tip ($)")
plt.legend()
plt.title("Locally Weighted Regression on Tips Dataset")
plt.show()
```

```
Predicted Tip for a total bill of $30: 225.10
```



In [ ]:

In [ ]: