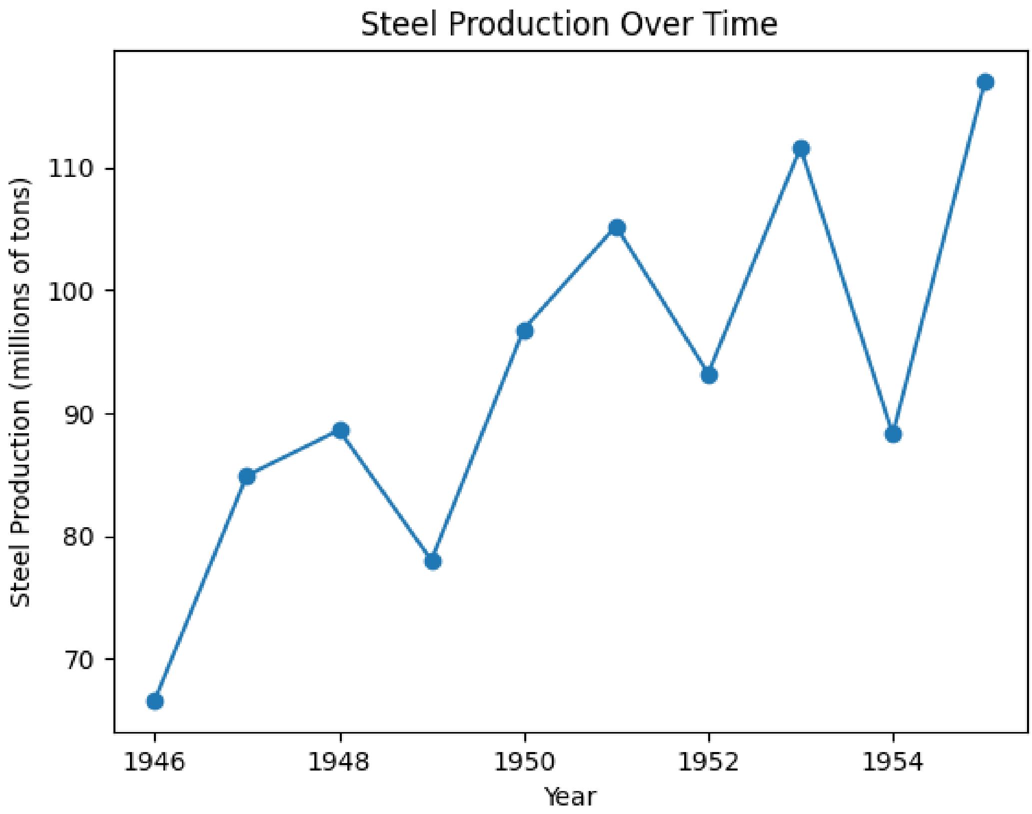
In [17]:

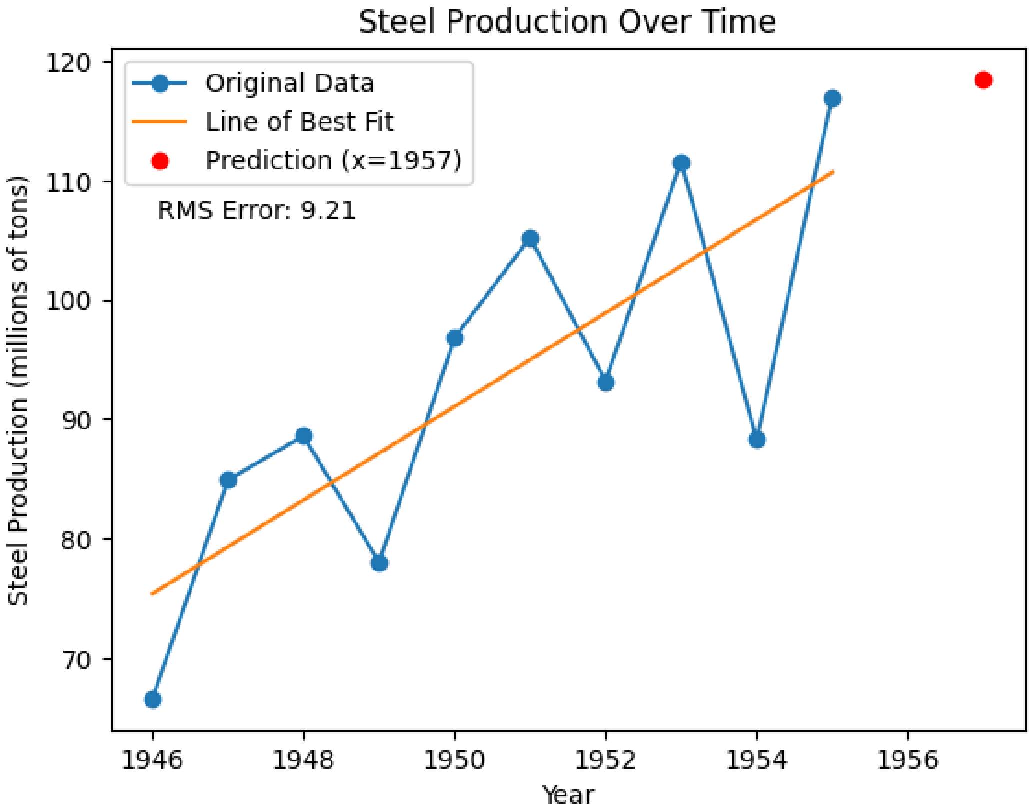
```
import numpy as np
import matplotlib.pyplot as plt

steel = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0]

years = np.arange(1946, 1956, 1)

plt.plot(years, steel, 'o-')
plt.xlabel('Year')
plt.ylabel('Steel Production (millions of tons)')
plt.title('Steel Production Over Time')
plt.show()
```



In [18]:

```
import numpy as np
import matplotlib.pyplot as plt

steel = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0]
years = np.arange(1946, 1956, 1)

# Perform linear curve fit
coefficients = np.polyfit(years, steel, 1)
poly = np.poly1d(coefficients)

# Generate the line of best fit
line_of_best_fit = poly(years)

# Calculate the RMS error
error = np.sqrt(np.mean((steel - line_of_best_fit) ** 2))

# Plot the original data and the line of best fit
plt.plot(years, steel, 'o-', label='Original Data')
plt.plot(years, line_of_best_fit, label='Line of Best Fit')
plt.xlabel('Year')
plt.ylabel('Steel Production (millions of tons)')
plt.title('Steel Production Over Time')
```

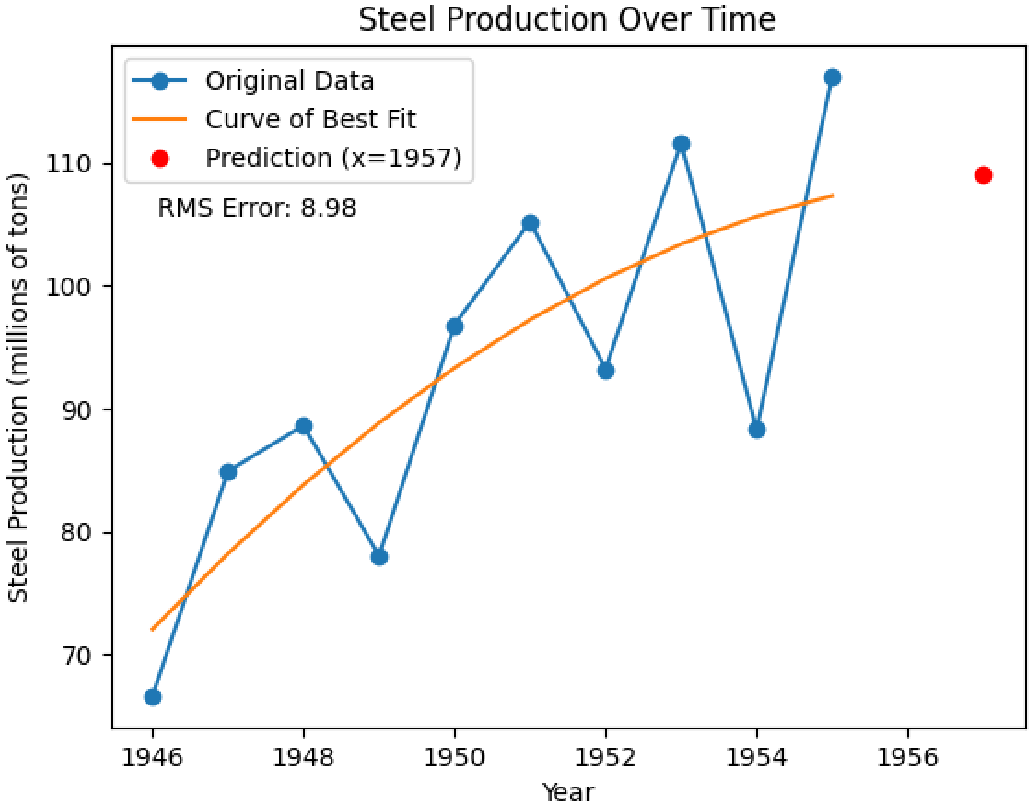
```
# Add the RMS error as a text annotation
plt.annotate(f'RMS Error: {error:.2f}', xy=(0.05, 0.75), xycoords='axes fraction')

# Add the prediction for x=1957
prediction = poly(1957)
plt.plot(1957, prediction, 'ro', label='Prediction (x=1957)')

plt.legend()
plt.show()
```



In [19]:

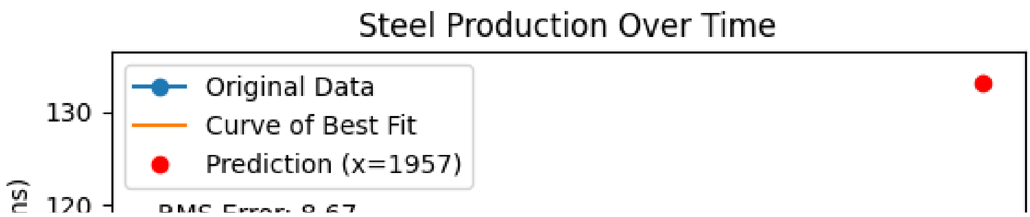
```
import numpy as np
import matplotlib.pyplot as plt

steel = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0]
years = np.arange(1946, 1956, 1)

# Perform quadratic curve fit
coefficients = np.polyfit(years, steel, 2)
poly = np.poly1d(coefficients)

# Generate the curve of best fit
curve_of_best_fit = poly(years)

# Calculate the RMS error
error = np.sqrt(np.mean((steel - curve_of_best_fit) ** 2))

# Plot the original data and the curve of best fit
plt.plot(years, steel, 'o-', label='Original Data')
plt.plot(years, curve_of_best_fit, label='Curve of Best Fit')
plt.xlabel('Year')
plt.ylabel('Steel Production (millions of tons)')
plt.title('Steel Production Over Time')

# Add the RMS error as a text annotation
plt.annotate(f'RMS Error: {error:.2f}', xy=(0.05, 0.75), xycoords='axes fraction')

# Add the prediction for x=1957
prediction = poly(1957)
plt.plot(1957, prediction, 'ro', label='Prediction (x=1957)')

plt.legend()
```

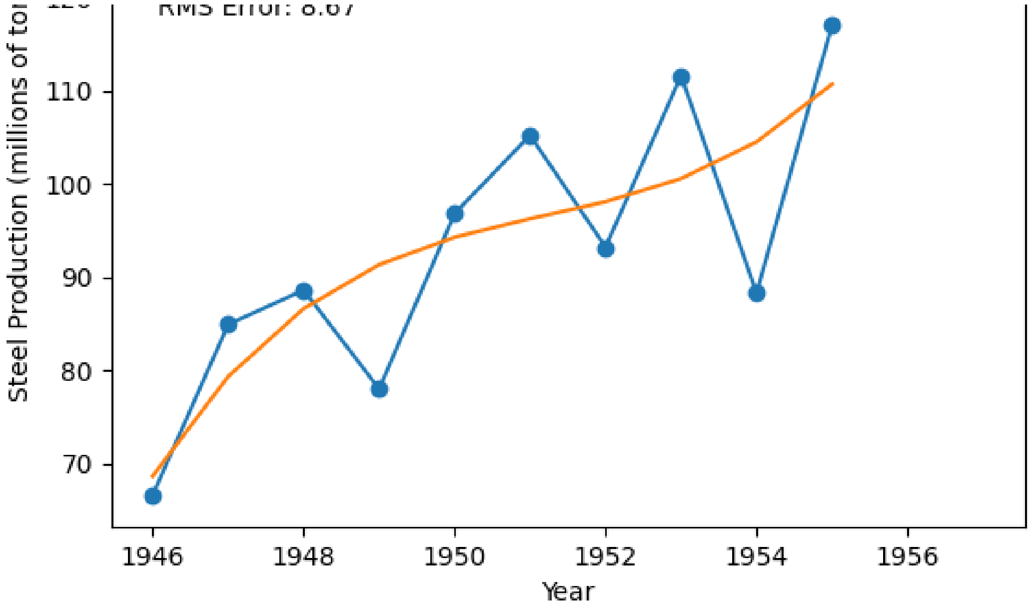
```
plt.show()
```



In [20]:

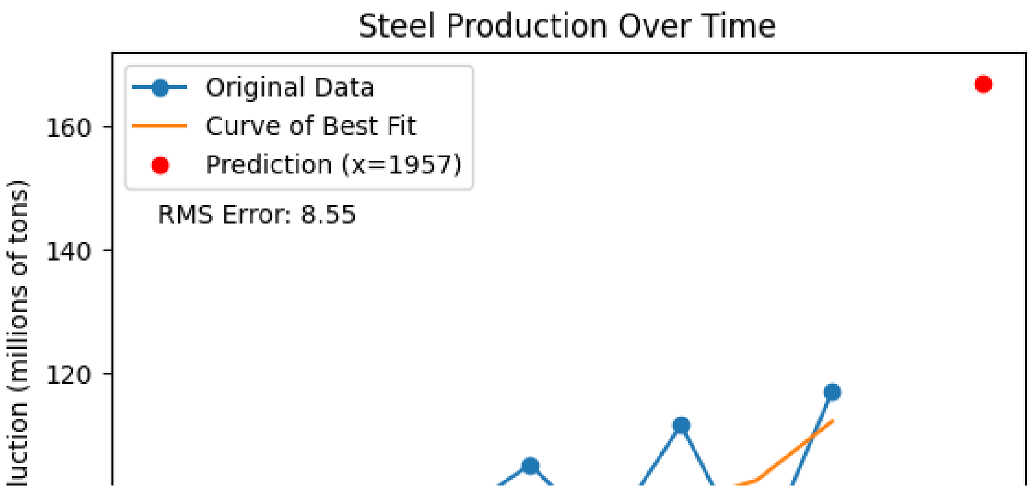
```
import numpy as np
import matplotlib.pyplot as plt

steel = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0]
years = np.arange(1946, 1956, 1)

# Perform quadratic curve fit
coefficients = np.polyfit(years, steel, 3)
poly = np.poly1d(coefficients)

# Generate the curve of best fit
curve_of_best_fit = poly(years)

# Calculate the RMS error
error = np.sqrt(np.mean((steel - curve_of_best_fit) ** 2))

# Plot the original data and the curve of best fit
plt.plot(years, steel, 'o-', label='Original Data')
plt.plot(years, curve_of_best_fit, label='Curve of Best Fit')
plt.xlabel('Year')
plt.ylabel('Steel Production (millions of tons)')
plt.title('Steel Production Over Time')

# Add the RMS error as a text annotation
plt.annotate(f'RMS Error: {error:.2f}', xy=(0.05, 0.75), xycoords='axes fraction')

# Add the prediction for x=1957
prediction = poly(1957)
plt.plot(1957, prediction, 'ro', label='Prediction (x=1957)')

plt.legend()
plt.show()
```

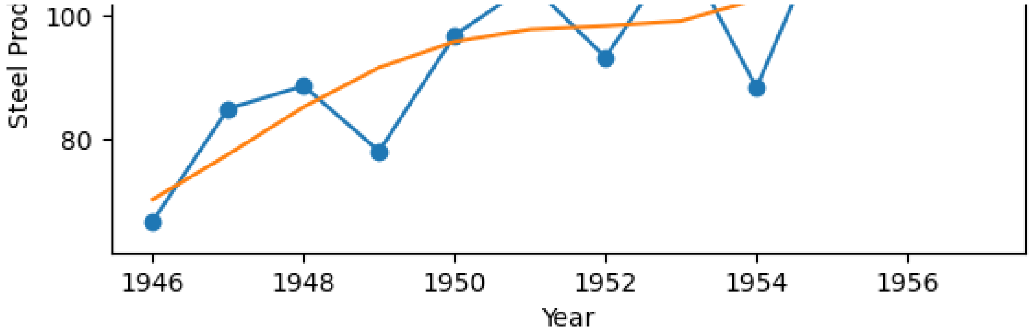




In [21]:

```
import numpy as np
import matplotlib.pyplot as plt

steel = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0]
years = np.arange(1946, 1956, 1)

# Perform quadratic curve fit
coefficients = np.polyfit(years, steel, 4)
poly = np.poly1d(coefficients)

# Generate the curve of best fit
curve_of_best_fit = poly(years)

# Calculate the RMS error
error = np.sqrt(np.mean((steel - curve_of_best_fit) ** 2))

# Plot the original data and the curve of best fit
plt.plot(years, steel, 'o-', label='Original Data')
plt.plot(years, curve_of_best_fit, label='Curve of Best Fit')
plt.xlabel('Year')
plt.ylabel('Steel Production (millions of tons)')
plt.title('Steel Production Over Time')

# Add the RMS error as a text annotation
plt.annotate(f'RMS Error: {error:.2f}', xy=(0.05, 0.75), xycoords='axes fraction')

# Add the prediction for x=1957
prediction = poly(1957)
plt.plot(1957, prediction, 'ro', label='Prediction (x=1957)')

plt.legend()
plt.show()
```





Steel Proc
100
80
1946
1948
1950
1952
1954
1956
Year