In [4]:

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

In [27]:

```
def f(x):
    return (x+3)**2
```

In [28]:

```
def df(x):
    return 2*(x+3)
```

In [29]:

```
def gradient_decent(start_x, learning_rate, num_iterations):
    x = start_x
    history_x= [x]
    for i in range(num_iterations):
        grad = df(x)
        x = x - learning_rate * grad
        history_x.append(x)
    return x,history_x
```

In [44]:

```
x = 2
learning_rate = 0.1
num_iterations = 50
min_x,history_x = gradient_decent(x, learning_rate, num_iterations)
print(min_x)
```

```
-2.9999286376153647
```

In [45]:

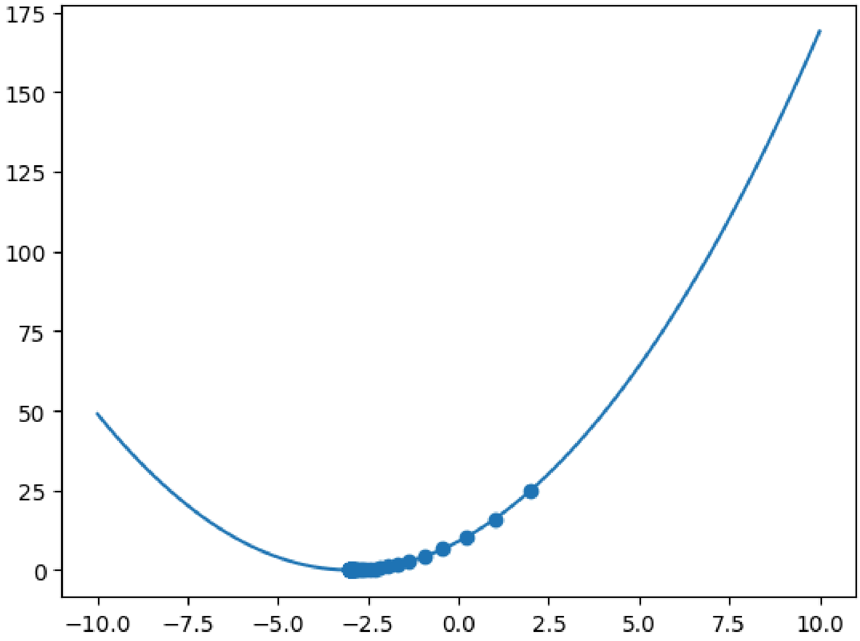
```
# plot graph of function, also of gradient descent

x = np.linspace(-10,10,100)
y = f(x)

plt.plot(x,y)

history_y = [f(i) for i in history_x]

plt.scatter(history_x,history_y)
plt.show()
```

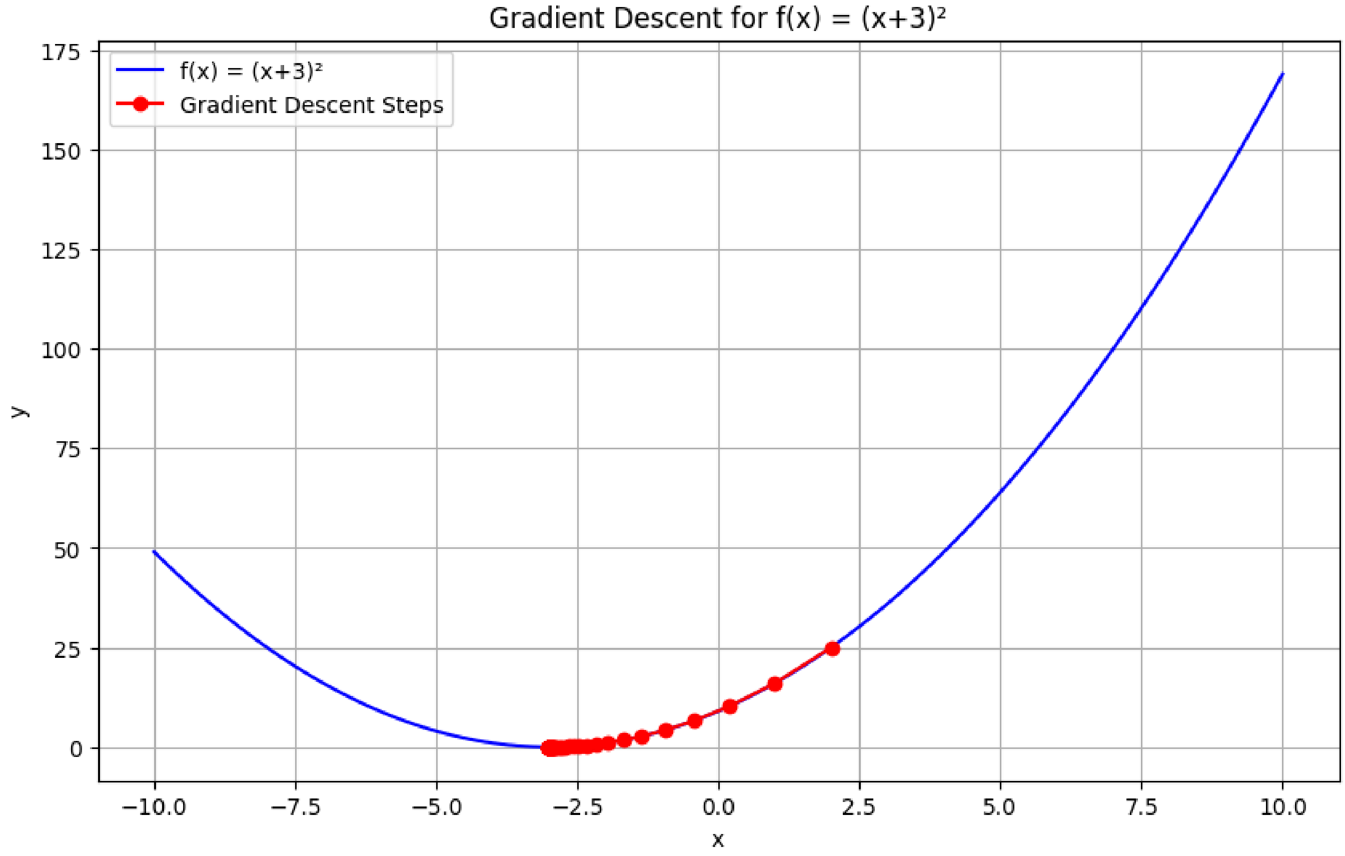


In [46]:

```
# Plot the function and gradient descent steps
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'b-', label='f(x) = (x+3)²')
plt.plot(history_x, history_y, 'ro-', label='Gradient Descent Steps')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gradient Descent for f(x) = (x+3)²')
plt.legend()
plt.grid(True)

plt.show()

print(f"Local minimum found at x = {min_x:.4f}")
print(f"f(x) at minimum = {f(min_x):.4f}")
```



```
Local minimum found at x = -2.9999
f(x) at minimum = 0.0000
```

In [ ]: