ass3ml

[1]:
```
# Importing necessary libraries
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
# Define the function and its derivative
def f(x):
    return (x + 3) ** 2

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

[3]:
```
# Gradient Descent Implementation
def gradient_descent(starting_point, learning_rate, num_iterations):
    x = starting_point
    x_history = [x]
    for _ in range(num_iterations):
        grad = gradient(x)
        x = x - learning_rate * grad
        x_history.append(x)
    return x, x_history
```

[4]:
```
# Parameters
starting_point = 2  # Starting point x=2
learning_rate = 0.1  # Learning rate
num_iterations = 30  # Number of iterations
```

[5]:
```
# Running Gradient Descent
minima, x_history = gradient_descent(starting_point, learning_rate,␣
 ↪num_iterations)
```

[6]:
```
# Results
print(f'Local minima found at: x = {minima}, y = {f(minima)}')
```

```
Local minima found at: x = -2.993810299803573, y = 3.83123885216492e-05
```

[7]:
```
# Plotting the function and the gradient descent path
x_values = np.linspace(-6, 0, 100)
```

```
y_values = f(x_values)
```

[8]:
```
plt.plot(x_values, y_values, label='y = (x + 3)^2')
plt.scatter(x_history, f(np.array(x_history)), color='red', label='Gradient␣
↪Descent Path')
plt.title('Gradient Descent on y = (x + 3)^2')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid()
plt.show()
```