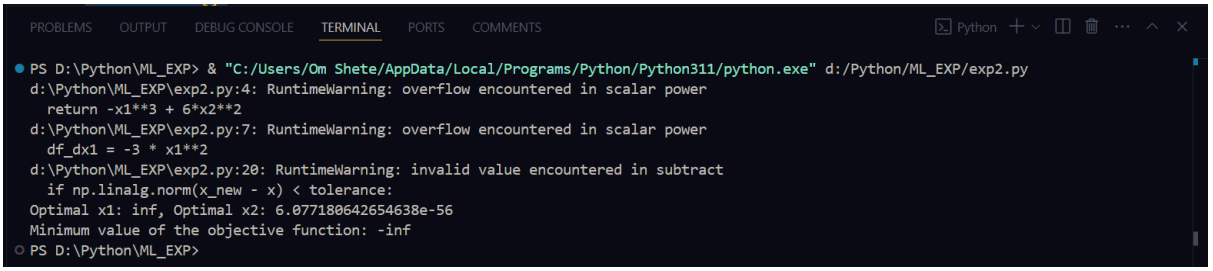
Code:

```
import numpy as np

def objective_function(x1, x2):
    return -x1**3 + 6*x2**2

def gradient(x1, x2):
    df_dx1 = -3 * x1**2
    df_dx2 = 12 * x2
    return np.array([df_dx1, df_dx2])

def gradient_descent(x_init, learning_rate=0.01, tolerance=1e-6,
max_iters=1000):
    x = np.array(x_init, dtype='float64')
    values = []

    for i in range(max_iters):
        grad = gradient(x[0], x[1])
        x_new = x - learning_rate * grad
        values.append(objective_function(x_new[0], x_new[1]))

        if np.linalg.norm(x_new - x) < tolerance:
            print(f'Converged in {i+1} iterations')
            break
        x = x_new

    return x, values

x_init = [2.0, 2.0]

optimal_x, values = gradient_descent(x_init, learning_rate=0.01)

print(f'Optimal x1: {optimal_x[0]}, Optimal x2: {optimal_x[1]}')

print(f'Minimum value of the objective function:
{objective_function(optimal_x[0], optimal_x[1])}')
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
Python
PS D:\Python\ML_EXP> & "C:/Users/Om Shete/AppData/Local/Programs/Python/Python311/python.exe" d:/Python/ML_EXP/exp2.py
d:\Python\ML_EXP\exp2.py:4: RuntimeWarning: overflow encountered in scalar power
return -x1**3 + 6*x2**2
d:\Python\ML_EXP\exp2.py:7: RuntimeWarning: overflow encountered in scalar power
df_dx1 = -3 * x1**2
d:\Python\ML_EXP\exp2.py:20: RuntimeWarning: invalid value encountered in subtract
if np.linalg.norm(x_new - x) < tolerance:
Optimal x1: inf, Optimal x2: 6.07718064265463 8e-56
Minimum value of the objective function: -inf
PS D:\Python\ML_EXP>