In [ ]:

```
#Name: Thokala Kavyasree
#Roll.No : 25201318
```

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
```

In [2]:

```
# Generate 100 random weights (0-100)
np.random.seed(42)
weights = np.random.uniform(0.000, 0.9999, 100) * 100
```

In [3]:

```
# Assign labels based on weight threshold
# Obese (1) if weight > 50, otherwise Not Obese (0)
labels = (weights > 50).astype(int)
```

In [4]:

```
#  Fit logistic regression
model = LogisticRegression()
model.fit(weights.reshape(-1, 1), labels)
```

Out[4]:

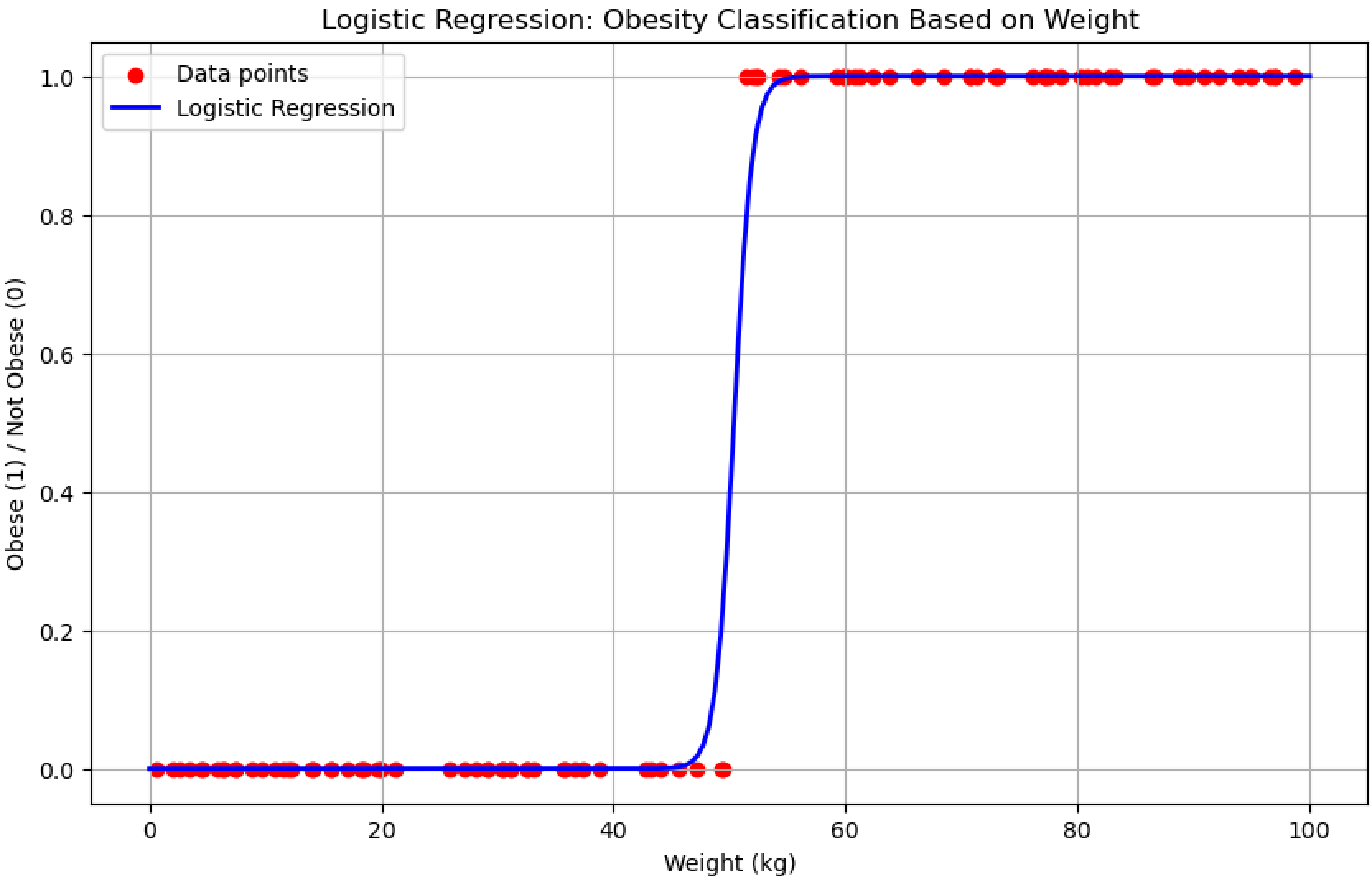
```
LogisticRegression()
```

In [5]:

```
#  Generate smooth weight range for plotting
weight_range = np.linspace(0, 100, 200).reshape(-1, 1)
pred_prob = model.predict_proba(weight_range)[:, 1]
```

In [6]:

```
#  Plot results
plt.figure(figsize=(10, 6))
plt.scatter(weights, labels, color='red', label='Data points')
plt.plot(weight_range, pred_prob, color='blue', linewidth=2, label='Logistic Regression')
plt.xlabel('Weight (kg)')
plt.ylabel('Obese (1) / Not Obese (0)')
plt.title('Logistic Regression: Obesity Classification Based on Weight')
plt.legend()
plt.grid(True)
plt.show()
```



In [7]:

```
#  Test predictions on given weight values
test_weight = pd.DataFrame({"weight": [10, 22, 27, 35, 70]})
predictions = model.predict(test_weight)

print("Test Predictions:")
for val, pred in zip(test_weight['weight'], predictions):
    print(f"Weight: {val}, Prediction: {'Obese' if pred == 1 else 'Not Obese'}")
```

```
Test Predictions:
Weight: 10, Prediction: Not Obese
Weight: 22, Prediction: Not Obese
Weight: 27, Prediction: Not Obese
Weight: 35, Prediction: Not Obese
Weight: 70, Prediction: Obese
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:443: UserWarning: X has feature names, but LogisticRegressio
n was fitted without feature names
  warnings.warn(
```

In [ ]: