In [24]:
```
import numpy as np
from qiskit.quantum_info import Statevector  # Corrected import
import matplotlib.pyplot as plt
from IPython.display import display, Latex
from qiskit import QuantumCircuit
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
ket0 = np.array([1, 0])
ket1 = np.array([0, 1])
```

In [5]:
```
#State |ψ⟩ = sqrt(5/9)|0⟩ + sqrt(4/9)|1⟩

superposition_state = np.sqrt(5/9) * ket0 + np.sqrt(4/9) * ket1
print(f"Superposition State : {superposition_state}")
```

```
Superposition State : [0.74535599 0.66666667]
```

In [6]:
```
state = Statevector(superposition_state)

if state.is_valid():
    print("The state is valid.")
else:
    print("The state is not valid.")
```

```
The state is valid.
```

In [12]:
```
display(state.draw('latex'))
```

$$\frac{\sqrt{5}}{3}|0\rangle+\frac{2}{3}|1\rangle$$

In [14]:
```
qc = QuantumCircuit(1)
qc.initialize(superposition_state, 0)
qc.measure_all()
```

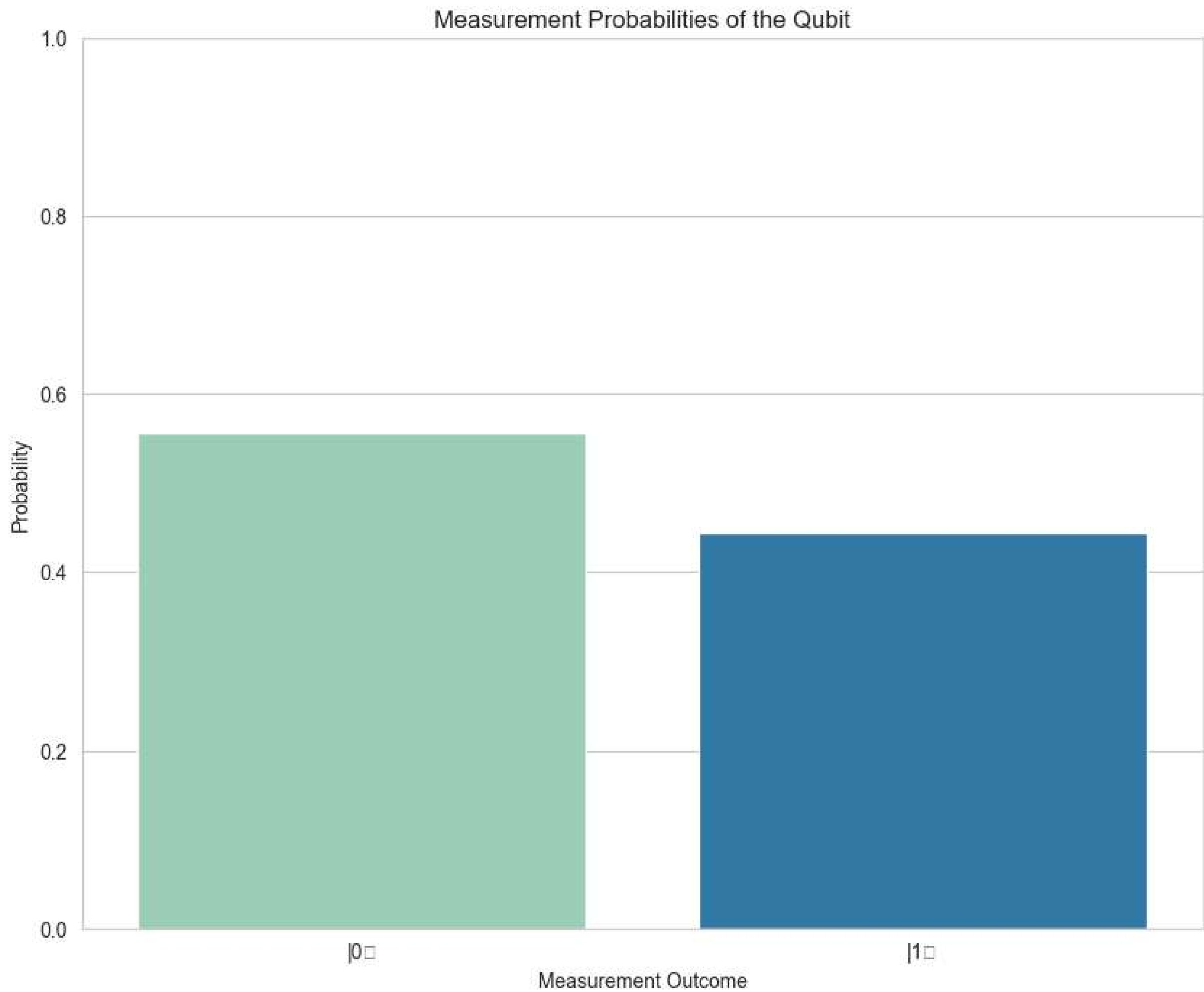
In [18]:
```
probabilities = np.abs(superposition_state) ** 2
measurement_outcomes = {'|0⟩': probabilities[0], '|1⟩': probabilities[1]}
print(f"Measurement Probabilities: {measurement_outcomes}")
```

```
Measurement Probabilities: {'|0⟩': 0.5555555555555556, '|1⟩': 0.4444444444444444}
```

In [25]:
```
sns.set_style('whitegrid')
color = sns.color_palette('YlGnBu', 2)

plt.figure(figsize=(10, 8))
sns.barplot(x=list(measurement_outcomes.keys()), y=list(measurement_outcomes.values
plt.xlabel('Measurement Outcome')
plt.ylabel('Probability')
plt.title('Measurement Probabilities of the Qubit')
```

```
plt.ylim(0, 1)  # Set y-axis limits to show probabilities
plt.show()
```



Inference for Quantum State Preparation and Measurement Probabilities

This code demonstrates the preparation and analysis of a quantum state using Qiskit. The primary goal is to create a quantum state in superposition and compute the theoretical measurement probabilities without utilizing the Aer module for simulation.

Key Components of the Code

1. **State Definition**:

   - The code defines two basis states, |0⟩ and |1⟩, as NumPy arrays.
   - An arbitrary superposition state |ψ⟩ is created using the coefficients (\sqrt{5/9}) and (\sqrt{4/9}).

2. **State Validity Check**:

   - The validity of the state vector is checked to ensure it is normalized (i.e., its norm should equal 1).
3. **Quantum Circuit Initialization**:
   - A quantum circuit is initialized with one qubit.
   - The qubit is prepared in the defined superposition state using the `initialize` method.
4. **Probability Calculation**:
   - Instead of executing the circuit on a simulator, the code directly computes the probabilities of measuring the qubit in the $|0\rangle$ or $|1\rangle$ states based on the state vector.
5. **Results Display**:
   - The measurement probabilities are displayed in a dictionary format, providing insight into the likelihood of measuring each state.
   - A bar chart visualizes these probabilities, making it easy to interpret the results.

Theoretical Measurement Outcomes

The computed probabilities for measuring the qubit in the $|0\rangle$ and $|1\rangle$ states are derived from the squares of the coefficients of the superposition state:

- Probability of $|0\rangle$: (\left(\sqrt{5/9}\right)^2 = \frac{5}{9} \approx 0.555)
- Probability of $|1\rangle$: (\left(\sqrt{4/9}\right)^2 = \frac{4}{9} \approx 0.444)

This indicates that when measuring the qubit, there is a 55.5% chance of observing it in the $|0\rangle$ state and a 44.4% chance of observing it in the $|1\rangle$ state.

Conclusion

The code effectively demonstrates the principles of quantum state preparation and probability measurement without the complexities of running a simulation. This approach is particularly useful for understanding quantum mechanics at a conceptual level while avoiding simulation tools.