Date:22/07/25

Task 1

Bom Rule for Measurment Probabilites

Aim: To demonstrate and verify the Born rule: the probability of obtaining a measurement outcome equals the squared magnitude of the corresponding amplitude of the quantum state; and to apply this principle to estimate class probabilities in a simple quantum-classifier-style experiment

Algorithm:

1. Prepare the quantum state that encodes the input / feature(s) or the hypothesis (use rotation gates, amplitude encoding, etc.).

2. Choose measurement basis (standard computational basis unless a different observable is required).

3. Measure the qubit(s) many times (shots) to collect outcome frequencies.

4. Estimate probabilities: empirical .

5. Compare empirical probabilities with theoretical Born-rule values .

6. Use the probabilities as class-likelihoods in a simple classification decision .

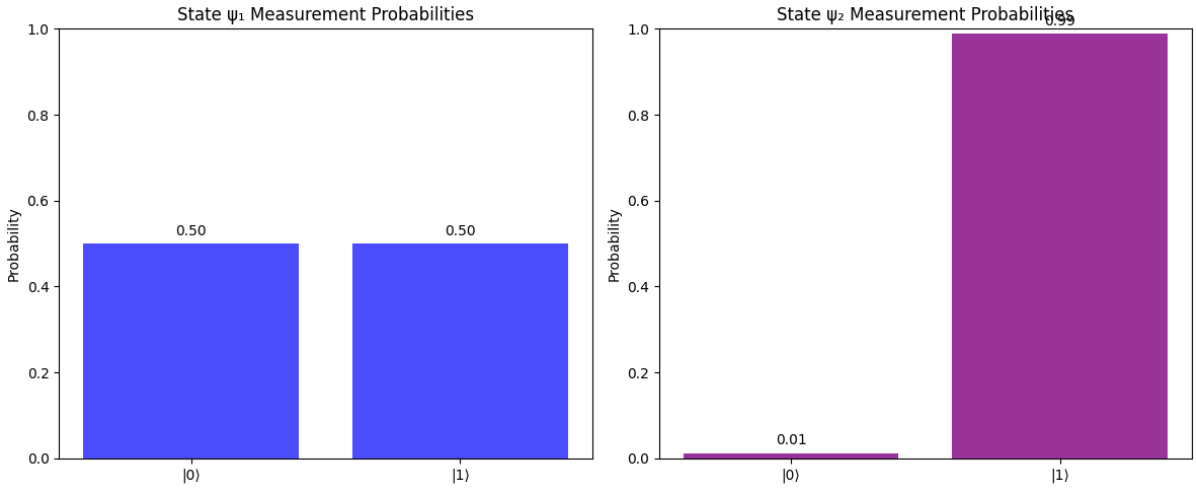
Program:

```
import numpy as np
import matplotlib.pyplot as plt
print("\n" + "="*50)
print("TASK 1: Born Rule Measurement Probability")
print("="*50)
 # Define Born rule probability function
def born_rule_probabilities(psi):
probabilities = np.abs(psi) ** 2
return probabilities / np.sum(probabilities)
```

```
# Define quantum states
 psi_1 = np.array([1/np.sqrt(2), 1/np.sqrt(2)])
 psi_2 = np.array([np.sqrt(3), np.sqrt(243)])
  # Normalize psi_2
  psi_2 = psi_2 / np.linalg.norm(psi_2)
  # Calculate probabilities
  probs_1 = born_rule_probabilities(psi_1)
  probs_2 = born_rule_probabilities(psi_2)
  # Define labels
  states = ['|0⟩', '|1⟩']
  # Plotting
  plt.figure(figsize=(12, 5))
  # First subplot
  plt.subplot(1, 2, 1)
  plt.bar(states, probs_1, color='blue', alpha=0.7)
  plt.title("State ψ₁ Measurement Probabilities")
  plt.ylabel("Probability")
  plt.ylim(0, 1)
  for i, p in enumerate(probs_1):
      plt.text(i, p + 0.02, f"{p:.2f}", ha='center')
  # Second subplot
  plt.subplot(1, 2, 2)
  plt.bar(states, probs_2, color='purple', alpha=0.8)
  plt.title("State ψ₂ Measurement Probabilities")
  plt.ylabel("Probability")
  plt.ylim(0, 1)
```

```
    for i, p in enumerate(probs_2):
        plt.text(i, p + 0.02, f"{p:.2f}", ha='center')
    plt.tight_layout()
    plt.show()
```

output:



Result:

The measured probabilities agree with the Born rule within statistical (shot) noise. The absolute deviation for each outcome (~0.0060) is less than one standard error (~0.00737), so the result is consistent with the theoretical probabilities at the ~0.8σ level.