TASK 1

Born Rule for Measurement Probabilities

Aim: To compute measurement probabilities of the quantum states using the born rule.

Algorithm:-

1) Define Quantum Superposition states

2) Apply Born rule to compute measurement of the probabilities.

8) Normalize probabilities"

4) visualize Jusuuts usi using ba bar chart.

Mathematical Model:

SOFTWARE FRAME WORK

\*Qiskit: It is an open-source frame work for the working with quantum circuits, simulations, and hardware which is developed by IBM

\* project Q: It is an open source software platform for Quantum computing developed by ETH Zurich.

\* penny Lane! It is the python library for the quantum machine learning.

\*Ten Sorflow Quantum (TFQ): It is a frame work that integrates Quantum Computation algorithm & the Circuit simulation with tensor flow.

\* cirg: It is a python frame work for designing, on Simulating, and running Quantum algorithms both hardware and simulatons, developed by the Google

\*py Quil:- It is a python library developed by Rigette computing for Quantum propgramming.

\* Amazon Braket: It is a fully managed Service that providus for quantum Computing Aws Clout a developmed environmen,Code:

import numpy as np

import matplotlib.pyplot as plt

print("\n" + "="\*50)

print("TASK 1: BORN RULE - MEASUREMENT PROBABILITIES")

print("="\*50)

def born\_rule\_probabilities(psi):

    probabilities = np.abs(psi)\*\*2

    return probabilities / np.sum(probabilities)

psi\_1 = np.array([1/np.sqrt(2), 1/np.sqrt(2)])

psi\_2 = np.array([1/np.sqrt(3), np.sqrt(2/3)])

print("Superposition state 1: |ψ₁⟩ =", psi\_1)

print("Measurement probabilities:", born\_rule\_probabilities(psi\_1))

print("\nSuperposition state 2: |ψ₂⟩ =", psi\_2)

print("Measurement probabilities:", born\_rule\_probabilities(psi\_2))

states = ['|0⟩', '|1⟩']

probs\_1 = born\_rule\_probabilities(psi\_1)

probs\_2 = born\_rule\_probabilities(psi\_2)

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)

plt.bar(states, probs\_1, color='blue', alpha=0.7)

plt.title('State |ψ₁⟩ Probabilities')

plt.ylabel('Probability')

plt.subplot(1, 2, 2)

plt.bar(states, probs\_2, color='red', alpha=0.7)

plt.title('State |ψ₂⟩ Probabilities')

plt.ylabel('Probability')

plt.tight\_layout()

plt.show()

OUTPUT:

