```
# Install PennyLane
# !pip install pennylane --quiet

import pennylane as qml
from pennylane import numpy as np
import matplotlib.pyplot as plt
from collections import Counter

# Step 1: Define a quantum device with 1 wire (qubit) and 1000 shots
dev = qml.device("default.qubit", wires=1, shots=1000)

# Step 2: Define the quantum function (quantum node)
@qml.qnode(dev)
def quantum_coin_flip():
    qml.Hadamard(wires=0)   # Put qubit into equal superposition
    return qml.sample(qml.PauliZ(0))  # Sample measurement outcomes

# Step 3: Run the circuit
samples = quantum_coin_flip()

# Step 4: Convert PauliZ outcomes (+1, -1) to coin outcomes (0, 1)
coin_results = [0 if s == 1 else 1 for s in samples]  # +1 -> 0 (heads), -1 -> 1 (tails)
counts = dict(Counter(coin_results))

# Step 5: Print and plot results
print("Quantum Coin Flip Results (0=heads, 1=tails):", counts)

# Plot
plt.bar(['0', '1'], [counts.get(0, 0), counts.get(1, 0)], color=['#1f77b4', '#ff7f0e'])
plt.title("Quantum Coin Flip Simulation (PennyLane)")
plt.xlabel("Outcome")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()
```