Title: Single-Qubit Superposition Simulation (Qiskit)
Author: Priyanshu Singh

Abstract (3–4 sentences)
I built a short Python program using IBM's Qiskit to demonstrate **quantum superposition** on a single qubit. The circuit applies a **Hadamard (H)** gate to prepare an equal superposition and measures the qubit **1,000 times** on a simulator. The resulting **histogram** shows roughly equal counts of 0 and 1, illustrating probabilistic **state collapse** at measurement. Through this project I learned core Qiskit workflows (AerSimulator, transpile), how to interpret shot statistics, and how to maintain a clean Python environment.

Background (Concept)
Classical bits are always 0 or 1. A **qubit** can exist in a **superposition**—a blend of 0 and 1 - until measurement. The **Hadamard** gate creates an equal superposition; repeated measurements reveal a ~50/50 distribution. Qiskit lets us model and visualize that process.

Method (What I built)

- Built a circuit with **1 qubit / 1 classical bit**.
- Applied **H** to create superposition; measured in the **Z basis**.
- Simulated **1,000 shots** with **AerSimulator** and plotted a histogram.

Results (What the histogram shows)

- Outcomes cluster near **50% '0'** and **50% '1'**; small differences are normal sampling noise.
- The plot is a direct picture of quantum **probabilistic measurement**, contrasting with classical determinism.

What I Learned

- Physics: superposition, basis measurement, and probabilistic outcomes.
- Tools: modern Qiskit API (AerSimulator, transpile), Matplotlib.
- Practice: version management (Python 3.13), debugging imports, clean environment setup.