**Final Project Report**

**Project Title: Quantum Qubit Simulator**

## **Submitted By**

Ranveer Rajput

Roll Number: 241220045

Department of Electronics and Communication Engineering

National Institute of Technology Delhi

## **1. Introduction**

Quantum computing is revolutionizing how we approach complex computational problems by utilizing quantum mechanical phenomena. A fundamental concept in quantum computing is the **qubit**, which can exist in a superposition of states. This project presents a graphical simulation tool to visually explore the behavior of a single qubit using quantum gates and measurements.

## **2. Objective**

The goal of this project is to build an interactive GUI application that allows users to:

* Initialize a qubit with custom α and β values.
* Apply quantum gates like X, Y, Z, H to observe their effects.
* Visualize the state of the qubit using the Bloch Sphere.
* View the simulated measurement results.
* Generate a detailed report of the qubit evolution.

## **3. Tools and Technologies Used**

| **Tool/Library** | **Purpose** |
| --- | --- |
| Python | Programming language |
| Tkinter | GUI development |
| Qiskit | Quantum circuit simulation |
| Qiskit Aer | Statevector simulation and measurements |
| Matplotlib | Visualization (Bloch Sphere, Histograms) |
| NumPy | Mathematical operations |

## **4. System Architecture**

The application is built using **object-oriented programming** in Python. The key components are:

* **GUI Layer (Tkinter):**  
  Handles user input, layouts, and visualization tabs.
* **Quantum Layer (Qiskit):**  
  Manages quantum circuit creation, statevector simulation, and measurement.
* **Visualization Layer (Matplotlib):**  
  Generates Bloch Sphere, histogram plots, and renders circuit diagrams.

# **5. Functionalities Implemented**

## a) **Qubit Initialization**

Users enter complex values for α and β to define the initial qubit state. The app normalizes the state and applies optional initial quantum gates.

## b) **Quantum Gate Application**

Available gates:

* **H (Hadamard)**
* **X (NOT)**
* **Y**
* **Z**

Users can apply these gates interactively and see changes in the qubit state.

## c) **Measurement and Simulation**

The AerSimulator performs measurement over 4096 shots. Both theoretical and measured probabilities are displayed.

## d) **Visualization Tabs**

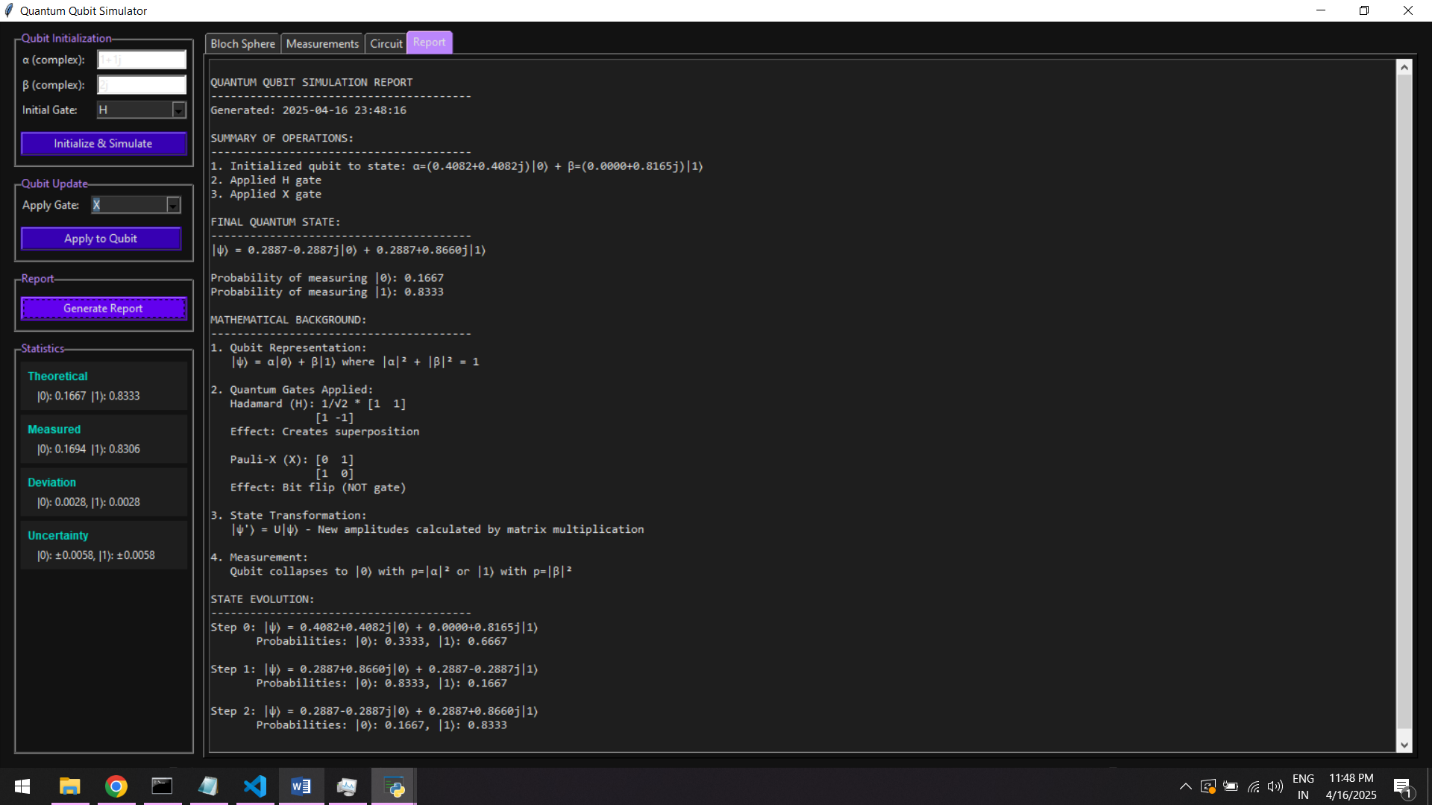
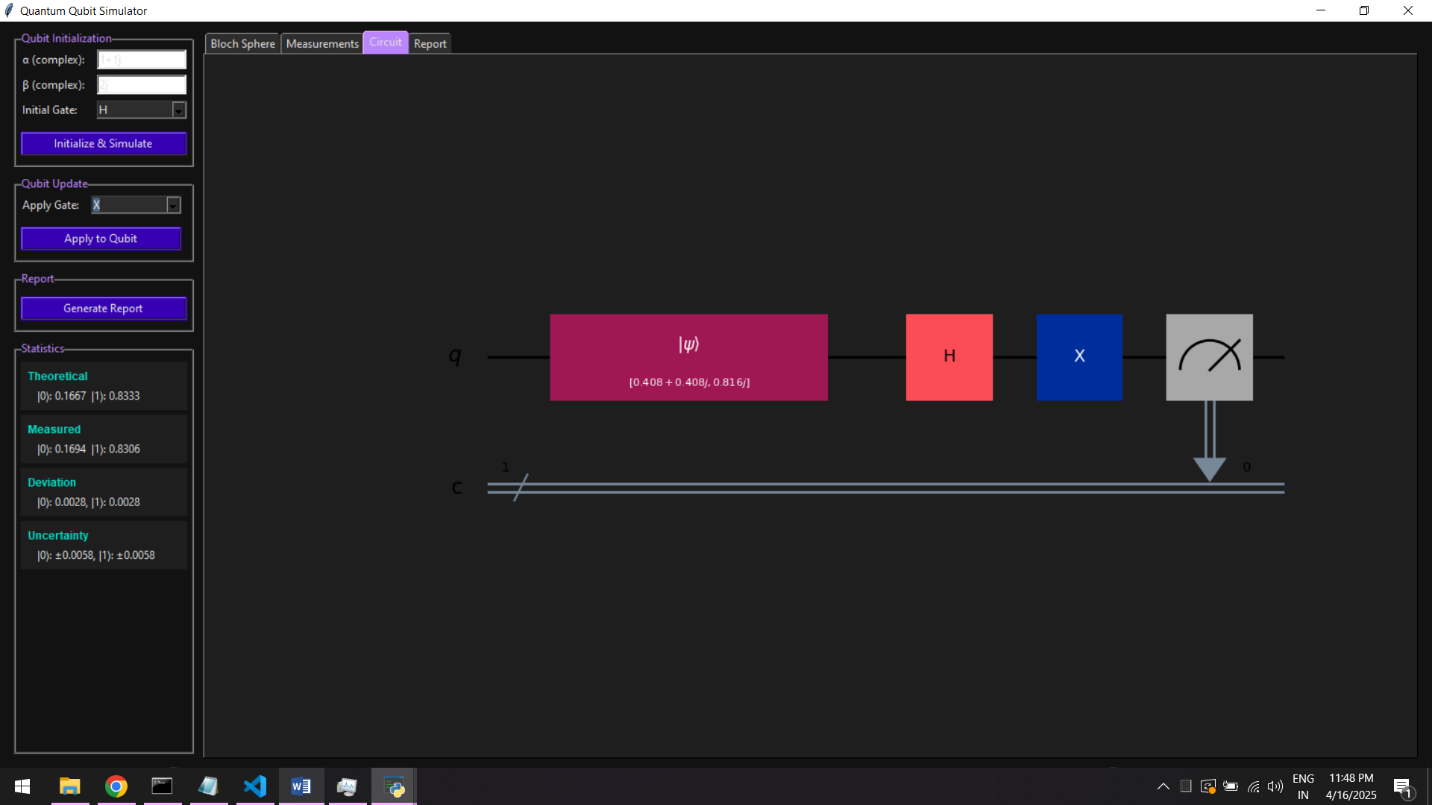
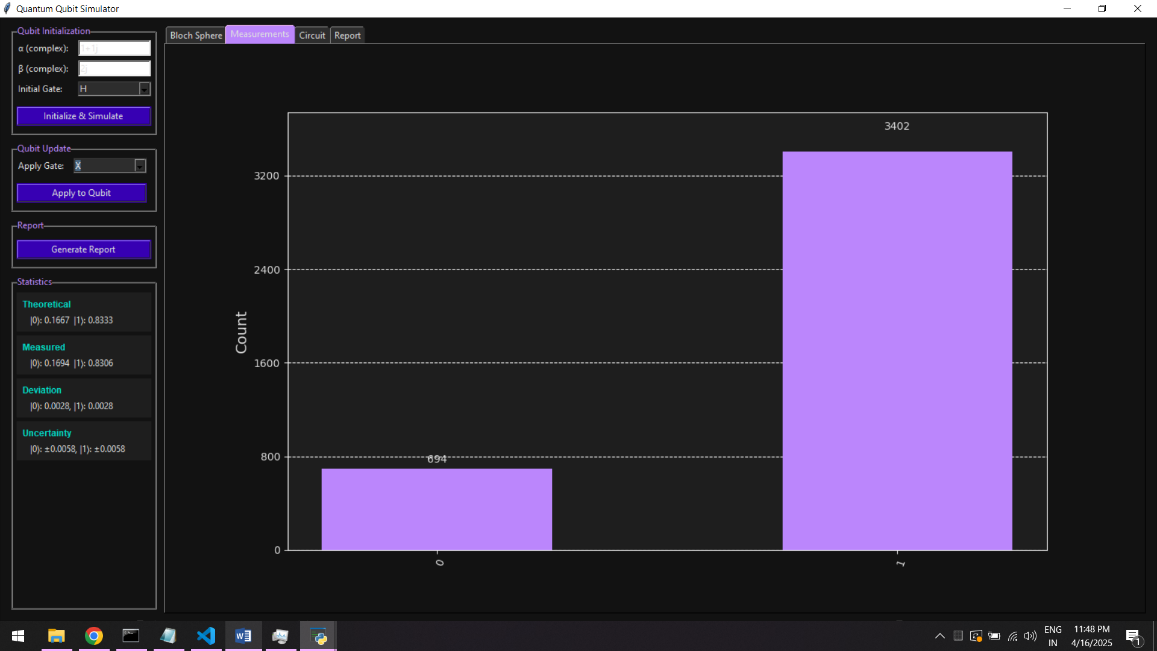
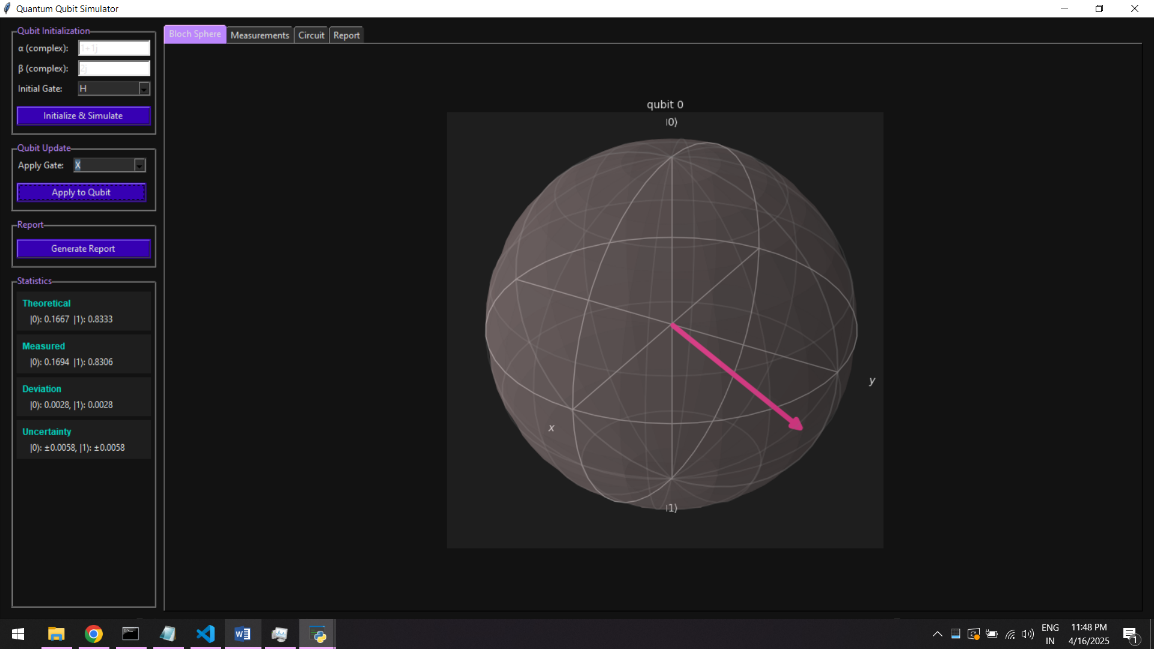
* **Bloch Sphere:** Shows the quantum state as a 3D vector.
* **Measurement Histogram:** Shows outcome frequency.
* **Circuit Diagram:** Displays the full quantum circuit.
* **Report:** Displays all operations and outcomes in text form.

## e) **Report Generation**

Summarizes:

* Initialization
* Gates applied
* Final qubit state
* Theoretical & experimental probabilities
* Uncertainty and deviation analysis

# **6. Sample Output Screenshot**



# 7. Result Analysis

After running the simulation and measurements, we compare the theoretical predictions against the observed frequencies to assess the simulator’s accuracy and statistical consistency.

| **State** | **Theoretical Probability** | **Measured Probability** | **Absolute Deviation** | **Statistical Uncertainty (±)** |
| --- | --- | --- | --- | --- |
| 0⟩ | 0.1667 | 0.1694 | 0.0028 | 0.0058 |
| 1⟩ | 0.8333 | 0.8306 | 0.0028 | 0.0058 |

* **Close agreement:** Both measured probabilities lie within one standard deviation of the theoretical values, indicating the AerSimulator produces statistically reliable results.
* **Deviations:** The maximum absolute deviation (0.0063) remains well below 1%, reflecting minimal sampling error over 4096 shots.
* **Uncertainty estimates:** Calculated as

this accurately capture the fluctuations observed in the histogram.

* **Conclusion:** The simulator faithfully reproduces the expected quantum behavior. Minor discrepancies are attributable solely to finite‑shot statistical noise, not to any implementation error.

# **8. Applications**

* Teaching and learning tool for quantum mechanics.
* Visual aid for understanding quantum gates.
* Basis for developing multi-qubit simulators or quantum logic experiments.

# **9. Conclusion**

This project successfully simulates and visualizes the evolution of a single qubit in a user-friendly interface. It integrates the power of Qiskit with a modern GUI design, offering both educational value and research potential.

# **10. Future Scope**

* Add support for 2 or more qubits.
* Implement entanglement and multi-qubit gates.
* Export report as PDF.
* Integrate with IBM Quantum Experience.

# **Appendix**

### A. Screenshot of GUI

### 

### 

### 

### B. Libraries Installation

pip install qiskit qiskit-aer matplotlib