Amaravati Quantum Valley Hackathon 2025

An Innovation Research Outcome Paper

Title of the Use Case:

Quantum Random Number Generator (QRNG) with integrated Randomness Analysis

Author(s) and Affiliation:

- **Team: SKY HIGH CODERS Team**
- **Team Members:** P. Charith, B. Yaswitha, G. Sree Lakshmi, B. Venkatesh, P. Charan Teja
- **Institute Name:** Shri Gnanambica Degree College (Autonomous), Madanapalle.
- **Team Contact Details:**

1. Email: chakri9701@gmail.com
   Phone: +91 9701859929
2. Email: boggalayaswithaboggala@gmail.com
   Phone: +91 6304024626
3. Email: sreelakshmi2484@gmail.com
   Phone: +91 9494068956
4. Email: balinenivenkatesnaidu@gmail.com
   Phone: +91 9182588110

Abstract:

Random numbers are important for security, online transactions, games, and research. Normal random number generators (PRNGs) are based on formulas, so they can be guessed or hacked. Our project builds a Quantum Random Number Generator (QRNG) that uses the laws of quantum mechanics to create truly random numbers. We also added a system to check and prove the randomness using statistical tests. The project was developed using Qiskit (quantum toolkit), FastAPI (backend), and React (frontend dashboard). The key highlight is that our system not only generates random numbers but also shows their quality through live reports and graphs.

---

1. Introduction:

Random numbers are everywhere — from encrypting messages, securing passwords, running simulations, to ensuring fairness in games. But normal random number generators are not fully secure because they use algorithms that can be predicted. Quantum mechanics, however, is unpredictable by nature. Measuring a quantum state always gives a random outcome. This project uses that principle to generate random numbers, then tests and shows their quality through a web-based platform.

2. **Problem Statement:**

- Problem: Current random number generators can be predicted or biased.
- Why it matters: In cybersecurity and banking, predictable randomness means higher chances of being hacked.
- Who benefits: Banks, Defence, IoT devices, cloud services, and researchers who need strong security and reliable simulations.

---

3. Literature Review:

- PRNGs (software-based) → fast but predictable.
- Hardware RNGs → better, but can still show bias or faults.
- Existing QRNGs → focus only on generating numbers, but rarely check or show their randomness quality.
- Gap: A complete solution that combines quantum random generation + real-time testing + user dashboard.

4. Methodology and Approach:

- Idea: Generate random numbers using quantum circuits, then run standard tests to check their quality.
- Tools Used:
  - Qiskit for quantum circuits
  - FastAPI for backend services
  - React for frontend visualization
- How it works:
  1. Quantum circuits run and create random bitstrings.
  2. Backend runs randomness tests (like frequency and runs test).
  3. Frontend shows results in graphs and reports.

---

5. Prototype Development:

- Type: Software prototype (works through cloud + web).
- Key Parts:
  - Quantum circuits (Hadamard gates for randomness).
  - APIs to collect numbers and test them.

    - React dashboard to display charts and test results.
    - Here's the Google Drive Link to see the demo UI

      https://drive.google.com/drive/folders/1cXsXRG9KKBr2b7blT94wPSydSLUXLL6J?usp=sharing
- Demo: The system shows random numbers, their distribution, and whether they pass randomness checks.

---

6. Implementation:

- Quantum Circuits: Created with Qiskit (superposition → measurement → random bits).
- Tests Used: Frequency test, runs test, chi-square test, entropy calculation.
- Challenges: Real hardware access is limited, so we used both simulators and IBM Quantum cloud.

---

7. Results and Outcomes:

- Generated thousands of random bits successfully.
- Most tests proved they were unbiased and close to perfect randomness.

- The dashboard made it easy to understand results visually.
- Our QRNG performed better than standard PRNGs in terms of unpredictability.

---

8. Innovation and Novelty:

- Unlike others, our QRNG includes live randomness validation and visualization.
- It connects research with practical, user-friendly applications.
- It could lead to future patents in secure randomness verification systems.

---

9. Use Case Applications:

- Cybersecurity: Stronger encryption keys.
- IoT and Cloud: Safer authentication.
- Gaming: Fair play and unpredictability.
- Research: Reliable results for simulations and models.

10. Limitations and Future Work:

- Right now, we depend on IBM Quantum cloud, which sometimes has delays.
- Next steps:
  - Build hardware-based QRNG.
  - Use AI to improve randomness testing.
  - Deploy at enterprise level for real-world use.

---

11. Conclusion:

Our project proves that quantum mechanics can generate secure, truly random numbers. By adding statistical tests and visual reports, we make randomness transparent and reliable. This work opens the door to future products that can strengthen global cybersecurity systems.

---

12. Acknowledgements:

We thank Amaravati Quantum Valley Hackathon organizers, mentors, and IBM Quantum for their help and resources.

13. References:

- NIST SP 800-22 Statistical Test Suite
- Qiskit Documentation – https://qiskit.org
- Research articles on “Quantum Random Number Generation” (Nature, IEEE)
- GitHub projects on randomness extractors and QRNG

---

14. GitHub Link:

- CharithPulaganti/QRNG: Quantum Random Number Generator

---

15. A SPECIAL THANKS TO EVENT ORGANISERS: