Module 1: Quantum Secuirty Foundations

Task: "Quantum Threats"

Research Component:

Investigate Shor's Algorithm and its specific application to the RSA-2048 prime factorization problem. Compare the classical time-complexity (number of operations) required to break it versus the quantum time-complexity.

Coding Component:

Use Python (NumPy/Qiskit) to simulate a small-scale Shor's algorithm for factoring the number 15 or 21.

Goal:

Visualize how quantum period-finding can find factors exponentially faster than classical trial-and-error.

Module 2: Quantum Communication (QC)

Task: "Securing Quantum Channel"

Research Component:

Analyze the BB84 protocol and the role of the Quantum Bit Error Rate (QBER) in detecting eavesdroppers. Research how Quantum Temporal Authentication (QTA) prevents a "Man-in-the-Middle" attack by verifying arrival times at the nanosecond scale.

Coding/Simulation Component:

Using QuNetSim or NetSquid, build a simulation of a 3-node network (Alice, Bob, and Eve).

Goal:

Successfully transmit a key while measuring the exact QBER threshold at which Alice and Bob must abort the session due to Eve's interference.

Module 3: Post-Quantum Cryptography (PQC)

Task: "FIPS 203–206 Implementation"

Research Component:

Compare the key and signature sizes of the four finalized NIST standards: ML-KEM (FIPS 203), ML-DSA (FIPS 204), SLH-DSA (FIPS 205), and the new FN-DSA (FIPS 206).

Coding Component:

Use the liboqs (Open Quantum Safe) library in C or Python to perform a hybrid key exchange.

Goal:

Implement a script that combines a classical algorithm (like X25519) with a PQC algorithm (ML-KEM-768) and measure the performance latency compared to a purely classical exchange.

Module 4: QRNG (Quantum Random Number Generators)

Task: "True Entropy Testing"

Research Component:

Study the ISO/IEC 23837 standard for the security evaluation of quantum modules. Identify three different physical sources of quantum entropy (e.g., optical beam splitting vs. vacuum fluctuations).

Coding Component:

Obtain a sample dataset of random numbers (many public QRNG datasets are available) and write a script to run the NIST SP 800-22 statistical test suite (or others).

Goal:

Verify if the "quantumness" of the source provides a superior entropy profile compared to a standard classical pseudo-random number generator (PRNG).

Module 5: Quantum Strategy & Risk Assesment (QRA)

Task: "The Crypto Audit (CBOM)"

Research Component:

Apply Mosca's Theorem ($X+Y > Z$) to a hypothetical bank that stores customer health data for 30 years. Determine their "migration deadline" based on current estimates for a cryptographically relevant quantum computer.

cryptographically relevant quantum computer.

Project Component:

Build a basic Cryptographic Bill of Materials (CBOM) for a simple open-source web application (like a basic Python/Flask site).

Goal:

Identify every point where RSA or ECC is used (TLS, SSH, DB encryption) and propose a tiered migration plan: what gets replaced with PQC first, and what requires a QKD hardware link.