Ledger Development Roadmap

Phase 1 — Foundation (Local Prototype)

Goal: functional standalone audit ledger (no cloud)

Objectives:
1. Design data model (event structure: timestamp, actor, action, details, prev_hash, hash)
2. Implement minimal C core: compute and verify hash chain
3. Build SQLite or local PostgreSQL database
4. Create Flask API (add event, list events, verify chain)
5. Add CLI or web dashboard (basic display + verify button)
6. Test with simulated security events
7. Document all design choices for CS50 submission

Deliverables:
- Local Flask app using C core via ctypes
- Hash-chain working correctly
- Demo video for CS50
- README + technical report

Phase 2 — Security & Cloud Deployment (Azure Integration)

Goal: deploy online securely under Coreventra domain

Objectives:
1. Move database to Azure PostgreSQL
2. Deploy Flask API on Azure App Service
3. Configure DNS for ledger.coreventra.com and api.coreventra.com/ledger
4. Add HTTPS, JWT authentication, and role-based access (admin/auditor)
5. Add signature verification and encryption of sensitive log fields
6. Introduce scheduled verification job (daily chain check)

Deliverables:
- Online ledger demo via HTTPS
- Verified audit log chain visible on dashboard
- Secure API with authentication

Phase 3 — Post-Quantum Security Upgrade

Goal: strengthen cryptography for future-proof design

Objectives:
1. Replace SHA3 with post-quantum hybrid hash/signature scheme (Dilithium, Kyber)
2. Integrate key management (Azure Key Vault)

3. Add verification routine supporting PQ signatures
4. Document cryptographic design decisions

Deliverables:
- PQ-ready ledger core
- Hybrid signing and verification pipeline
- Updated documentation and demo video

Phase 4 — Integration with Coreventra & Network Lab

Goal: connect Ledger to the Coreventra ecosystem

Objectives:
1. Allow other Coreventra modules (Network Lab, Treasury, Chat) to send audit events via API
2. Create connectors (Python client, webhook, or syslog adapter)
3. Add dashboards for each system type (security, financial, operational)
4. Integrate with AI assistant to summarize logs or detect anomalies

Deliverables:
- Unified audit backbone across Coreventra services
- Flask dashboard with filters per module
- Optional AI assistant for event summary and verification reporting

Phase 5 — Stabilization & Showcase

Goal: finalize for portfolio and recruiter presentation

Objectives:
1. Polish interface and documentation
2. Record demo video: add, verify, detect tampering
3. Write a short whitepaper describing the design (CS50 + cybersecurity relevance)
4. Publish project on GitHub (sensitive keys excluded)
5. Prepare presentation slides for recruiters or teachers

Deliverables:
- Final GitHub repository with technical docs
- Online demo on coreventra.com domain
- Professional presentation material