PROJECT TITLE: DrugAuthChain: Decentralized Anti-Counterfeit Medicine Protocol

Project Overview

- **Team Name:** ALGAERITHM
- **Team Members:**
    - Chongtham Allen
    - James Khuraijam
    - Khuraijam Dijen

Submission Links

- **Round 1 Concept Video:** hackathon vid 2.mov

---

1. THE PROBLEM LANDSCAPE

- **Problem Statement:** The pharmaceutical supply chain is plagued by counterfeit medicines that endanger lives. The core point is the inability of consumers to verify the authenticity of a drug packet instantly and the ease with which counterfeiters replicate static QR codes.
- **Target Audience:** Pharmaceutical Manufacturers (who need to protect their brand) and End Consumers (who need to verify medicine safety).
- **The 'Why Now':** With the rise of complex supply chains, traditional physical tamper-proof seals are easily forged. We need a cryptographic solution where the verification process itself invalidates the code for future reuse.

2. PROPOSED SOLUTION & USP

- **Solution Overview:** We propose a hybrid system using **Polygon Amoy** for immutable transaction records and **Google ML Kit** for rapid scanning. Manufacturers "mint" batch records as digital assets. When a consumer scans a product, the system performs a "Burn on Scan," marking the digital asset as "Sold" on the blockchain.
- **Unique Selling Proposition (USP):** Unlike standard track-and-trace systems, our solution employs a **"Burn on Scan" mechanism**. Once a medicine is scanned and verified by a consumer, its status updates to "Sold" on-chain, preventing any subsequent counterfeiter from refilling and reselling the same package.

- **Core Logic:** The "magic" lies in the Smart Contract's state management.
    1. **Mint:** Manufacturer generates a hash. Status = ACTIVE.

2. **Verify:** Consumer scans. Smart Contract checks status.
3. **Burn:** If Active, return "Authentic" and set Status $\rightarrow$ Sold. If already Sold, return "Fake".

3. TECHNICAL ARCHITECTURE & STACK

- **System Workflow:**
  - **Input:** Manufacturer inputs Batch ID in Dashboard Mints record on Polygon Metadata (photos/docs) uploaded to IPFS.
  - **Process:** Consumer scans QR via Google ML Kit App extracts ID Ethers.js queries Smart Contract.
  - **Output:** Contract returns status. If "Fake", location is sent to Firebase for the Admin Heatmap.
- **Tech Stack:**
  - **Frontend: Flutter (Dart)** - Chosen for a single, high-performance codebase for both iOS and Android.
  - **Blockchain Network: Polygon Amoy Testnet** - Selected for high throughput and low gas fees to ensure economic viability.
  - **Smart Contracts: Solidity** (Logic) & **Hardhat** (Dev Environment) - For compiling, testing, and deploying secure contracts.
  - **Blockchain Interaction: Ethers.js** & **MetaMask** - Ethers.js bridges the frontend to the blockchain; MetaMask handles manufacturer authentication.
  - **Backend & DB: Firebase (Google)** - Manages user profiles and "Report" logs; Cloud Functions trigger alerts.
  - **Decentralized Storage: IPFS** - Stores heavy metadata off-chain to keep the network unclogged and tamper-proof.
  - **AI/Specialized Tools: Google ML Kit** - Provides robust, offline-capable QR code scanning even in low-light conditions.

4. KEY FEATURES & FUNCTIONALITIES

- **Feature 1 (Primary - Trust): "Burn on Scan" Verification.** The core utility that validates a drug and simultaneously invalidates the QR code to prevent reuse.
- **Feature 2 (UX - Feedback): Visual Authenticity Feedback.** Uses **Lottie** animations (Green checkmark for success, pulsing Red for warning) to give users immediate, language-agnostic results.
- **Feature 3 (Reliability - Admin): Counterfeit Heatmap.** An Admin Dashboard using **Leaflet.js** that visualizes global locations where "Fake" scans are reported, aiding in supply chain investigation.

5. IMPLEMENTATION ROADMAP

Phase 1: Qualifying Round

- **Focus:** Ideation, Architectural Planning, and Initial Prototyping.
- **Tasks:**

- Finalize the User Flow and set up the Flutter/Hardhat development environment.
- Write and deploy **Solidity Smart Contracts** with `registerMedicine` and `verifyMedicine` functions to the local blockchain.
- Build the **Manufacturer Dashboard** to input Batch IDs and "Mint" new records using Ethers.js.
- Set up **Firebase** for non-blockchain data storage.

Phase 2: Final Excellence Round

- **Focus:** Feature Completion, UI/UX Refinement, and Final Demo.
- **Tasks:**
  - **Scanner Integration:** Implement **Google ML Kit** to read QRs and trigger the "Burn on Scan" blockchain transaction.
  - **UI Polish:** Design result screens with **Lottie animations** and a "Detailed Info" view (expiry, factory location).
  - **Analytics:** Build the **Leaflet.js Admin Heatmap** to track fake scans.
  - **Testing:** Edge-case testing (blurry QRs, slow internet) and bug hunting.
  - **Demo Prep:** Create physical demo props (boxes with QR codes) and record the final "Full Loop" demo video.

6. IMPACT & SUSTAINABILITY

- **Social/Economic Impact:** This project directly saves lives by removing counterfeit drugs from the market and protects the revenue and brand reputation of legitimate manufacturers.
- **Scalability:** The use of **Polygon Amoy** allows for high-volume transactions with negligible costs , and **IPFS** ensures storage scales indefinitely without bloating the blockchain.
- **Risk & Mitigation:**
  - *Risk:* Connectivity issues in remote areas preventing verification.
  - *Mitigation:* Implementation of **Offline Verification** by pre-loading public keys, allowing the app to verify digital signatures even without 5G access.

7. REFERENCES

1. **Polygon Documentation:** For Amoy Testnet configuration and Ethers.js integration.
2. **Google ML Kit:** Documentation for on-device barcode scanning APIs.
3. **IPFS Docs:** For decentralized file storage implementation.
4. **Flutter/Dart:** Official documentation for cross-platform mobile development.