# Project Proposal

## 1. Title & One-Line Value Proposition

Trusted Traceability for Food Supply Chains — farm‑to‑shelf provenance and quick recall using Hyperledger Fabric.

## 2. Problem & Stakeholders

Food items can be contaminated or mislabelled and tracing them through current systems is slow and fragmented. This causes consumer health risks, costly recalls for retailers, and compliance headaches for regulators. If solved, farmers, distributors, retailers, regulators, and consumers all benefit from faster, auditable provenance and lower recall costs.  
  
Stakeholders: Farmers, logistics providers (distributors), retailers (e.g., Walmart-style), regulators (FDA/USDA), consumers.

## 3. Research Alignment

Theme: Supply-chain traceability (anti-counterfeit, QR/NFC proofs, role-based flows).  
This project builds on IBM’s Food Trust / Walmart use case by implementing a minimal, role-enforced provenance ledger and a consumer verification flow.

## Prior Art: Walmart + IBM Food Traceability Use Case

Walmart is one of the most cited real-world adopters of blockchain in food supply chains. Below are highlights of what they did:  
  
- In 2016, Walmart and IBM launched pilot projects for mangoes in the U.S. and pork in China, tracking produce from farm through processing, shipping, and retail.  
- Before blockchain, tracing a package of sliced mangoes took Walmart over 6 days, 18 hours, and 26 minutes. With IBM/Hyperledger, the same trace took only 2.2 seconds.  
- Walmart required leafy greens suppliers to adopt standardized traceability protocols using GS1 data models on the IBM Food Trust network.  
- In China, Walmart worked with JD.com, IBM, and Tsinghua University to trace pork products on blockchain for regulators and supply chain actors.  
- These pilots showed dramatic benefits (faster tracebacks, transparency) but also challenges (supplier onboarding, data integration, and standardization).

## 4. Platform & Rationale

Choice: Hyperledger Fabric.  
Why: Fabric supports multi-organization networks, endorsement policies, and private data collections — useful to model real supply-chain orgs where some commercial data must remain private while provenance is auditable.

## 5. MVP Features + Stretch

MVP (to demo by Week 10)  
  
- Batch lifecycle chaincode — create a batch asset (manufacturer/farm), update status on ship/receive, and close at sale. This enforces roles and produces on-chain events.  
- QR/NFC per batch — each batch has a QR code that maps to a chain asset; scans call a gateway API to fetch the on-chain path for consumer verification.  
- Consumer verification portal — a minimal web page where scanning a QR shows the batch history (timestamps, org signatures). Demo: scan → show path within 60 seconds.  
  
Stretch (if time):  
- IoT temperature stream & Alerts — simulated sensor stream; if temperature threshold breached, emit an Alert event and mark batch quarantine.  
- Analytics dashboard — simple KPIs (verifiability rate, breach rate) with light DP noise for aggregated metrics.

## 6. Architecture Sketch

Actors: Farmer → Distributor → Retailer → Consumer. Physical goods flow left→right.  
Chaincode/Smart Contracts: Manage batch lifecycle (CreateBatch, ShipBatch, ReceiveBatch, CloseBatch) and emit events for each state change.  
Backend/Gateway: API that validates requests (signatures/roles), writes transactions to Fabric, ingests QR scans and simulated sensor data, and serves the consumer portal.  
UI / Consumer Portal: Web page or mobile web that scans QR/NFC and shows provenance steps pulled from chain events.  
Data sources: Synthetic batch manifests and simulated IoT sensor streams for temperature.  
Identity & Roles: Each organization has an MSP identity in Fabric; the gateway enforces that only authorized roles call the relevant chaincode functions.  
  
On‑chain events (examples): BatchCreated, BatchShipped, BatchReceived, TemperatureAlert, BatchVerified.

## 7. Security & Privacy Requirements

We will use Fabric endorsement policies to require multiple org approvals for sensitive updates. Private business data (e.g., pricing, contract details) will live in Private Data Collections; only hashes go on-chain. Transactions require signed gateway submissions and input validation; the gateway will rate-limit public endpoints (e.g., QR verification) to prevent spam. For analytics, we will add small DP noise to aggregated KPIs to avoid leaking batch-level sensitive info.

## 8. Milestones (Weeks 6–14)

W6: Environment up (Fabric test network), basic folder/repo structure, skeleton chaincode, one unit test. (Goal: prove basic network and commit template.)  
W7: Vertical slice demo — UI → gateway → chaincode → event. (Goal: end-to-end path for one batch.)  
W8: Feature 1 complete — full batch lifecycle implemented + event logging. (Goal: create/ship/receive/close flows working.)  
W9: Feature 2 complete — QR generation and consumer verification UI; role-based auth enforced. (Goal: consumers can verify provenance.)  
W10: Threat model and three mitigations implemented (endorsement, PDC, input validation). (Goal: basic security posture.)  
W11: Test suite (≥10 tests) and basic performance metrics recorded. (Goal: reliability checks and simple perf numbers.)  
W12: Add stretch: IoT simulation and/or analytics dashboard; LLM-assisted basic audit scripts if possible. (Goal: show one stretch feature.)  
W13: Freeze, polish UI/UX, documentation, and runbook. (Goal: reproducible demo.)  
W14: Dry-run poster/pitch and final fixes. (Goal: presentation-ready.)

## 9. Team & Roles + Logistics

Mohan — Backend/API (gateway, QR endpoints, IoT ingestion).  
Dinesh — Smart Contract / Chaincode (Fabric chaincode & tests).  
Aasritha — Frontend / UX (consumer portal, QR scan UI).  
Srija — PM / Scrum + DevOps & Testing (network setup, CI, test orchestration).  
  
Weekly standup: Fridays, 7:00 PM (Central).  
Comms: Slack / Discord + GitHub Issues & Project board.  
Repo URL: (add repo link here once created)

## 10. Top Risks & Mitigations

Data realism → Mitigation: use synthetic batch manifests and a small sensor generator to simulate realistic flows.  
Policy complexity (endorsement rules) → Mitigation: start with a simple allow-list and incremental endorsement policies per operation.  
Chaincode bugs/time → Mitigation: test-first development, unit tests for each function, and event assertions in CI.

## References

Walmart Tech Blog: Blockchain in the food supply chain - https://tech.walmart.com/content/walmart-global-tech/en\_us/blog/post/blockchain-in-the-food-supply-chain.html

IBM Media: Walmart’s food safety solution using IBM Food Trust - https://mediacenter.ibm.com/media/Walmart%27s%2Bfood%2Bsafety%2Bsolution%2Busing%2BIBM%2BFood%2BTrust%2Bbuilt%2Bon%2Bthe%2BIBM%2BBlockchain%2BPlatform/1\_zwsrls30

TechCrunch: Walmart betting on blockchain to improve food safety - https://techcrunch.com/2018/09/24/walmart-is-betting-on-the-blockchain-to-improve-food-safety/

Academic Case: Food Traceability on Blockchain: Walmart’s Pork and Mango Pilots - https://jbba.scholasticahq.com/article/3712-food-traceability-on-blockchain-walmart-s-pork-and-mango-pilots-with-ibm/attachment/20459.pdf

PixelPlex Blog: How Walmart enhances food safety via blockchain - https://pixelplex.io/blog/walmart-strives-for-food-safety-using-blockchain/

PMC Review: Application of blockchain technology in shaping the future of food - https://pmc.ncbi.nlm.nih.gov/articles/PMC10020414/