Project Roadmap: Blockchain and AI-Enabled Income Traceability System (TRACIENT)

💡 Phase 0: Knowledge Prerequisites (Pre-Build)

Core Concepts to Learn

Concept	Why It's Important
Blockchain Basics	Understand blocks, consensus, public/private chain, immutability
Hyperledger Fabric	Framework for private blockchain, suited for gov/enterprise use
Go (Golang)	Primary language for writing smart contracts (chaincode) in Fabric
Chaincode Logic	How smart contracts enforce wage logging, BPL classification, etc.
AI/ML Models	Needed for income prediction, classification (BPL/APL), fraud detection
Data Privacy + ZKP	Ensure secure, non-leaky architecture for sensitive IDs like Aadhaar

---

🗜️ Phase 1: Foundation & Proof of Concept (Months 1-3)

Goals:

- Set up initial Hyperledger Fabric blockchain network
- Write basic Go chaincode to log wage data
- Build a working frontend + backend demo (PoC)

Key Tasks:

1. **Team Setup**

2. Roles: Blockchain Developer (Go), AI/ML Engineer, Frontend Developer, Backend Developer

3. Tools: GitHub, Docker, VS Code, Go 1.20+, Python 3.9+, Fabric binaries

4. **Dev Environment Setup**

5. Folder structure: `blockchain/`, `ai-model/`, `backend/`, `frontend/`

6. Docker setup for Fabric

7. **Architecture Design**

8. Finalize system flow: Wage -> Chaincode -> AI -> Dashboard

9. Define schemas: `workerID`, `txnID`, `amount`, `employerID`, `sector`, etc.

10. **Fabric PoC Network**

11. One-org setup via Docker Compose

12. Go chaincode:

```go
func recordWage(workerID string, employerID string, amount float64,
timestamp string)
func queryWageHistory(workerID string)
```

13. **Simple API + Frontend**

14. Node.js/Go backend to call chaincode

15. React UI: form to log wage, wage history viewer

⚙️Phase 2: MVP Development (Months 4-7)

Goals:

- Full blockchain + AI-backed MVP
- Support for multi-org, user roles, dashboards

Key Tasks:

1. **Multi-org Fabric Network**

2. Setup orgs: Government, Employers, NGOs

3. Add identity management (CA, certificates)

4. **Advanced Chaincode in Go**

5. Access control: Only authorized employers log wages

6. New functions:

```go
func updateBPLStatus(workerID string, isBPL bool)
```

7. Merkle proof for verification

8. **AI/ML Model Integration**

9. Simulated wage dataset

10. ML Models:

    - Linear Regression (income estimation)
    - Random Forest (BPL classification)
    - Isolation Forest (anomaly detection)

11. **Backend API Layer**

12. Trigger ML models on new wage entries

13. Chaincode call to update eligibility flag

14. **UI Panels**

15. Employer Panel

16. Worker Panel (view income, history)
17. Government Dashboard

🔎Phase 3: Integration & Security (Months 8-10)

Goals:

- Finalize data flow, integrate real-world systems, and secure entire system

Key Tasks:

1. **Integration**

2. Simulated APIs: Aadhaar eKYC, UPI sandbox, Exotel/Twilio (IVR/SMS)

3. **Testing**

4. Unit: Chaincode + API

5. Integration: End-to-end from wage entry → dashboard

6. Performance: 10k txns stress test

7. **Security Hardening**

8. Anonymize Aadhaar/PAN using SHA256

9. Add Zero-Knowledge Proofs (ZKPs)
10. Implement RBAC, encryption (TLS), penetration testing

💸Phase 4: Pilot Deployment (Month 11-12)

Goals:

- Launch system for live users in limited area

Key Tasks:

1. **Pilot Site Selection**

2. 1 Panchayat, construction group, or small NGO

3. Onboard 50–100 workers + employers

4. **Cloud Deployment**

5. AWS/GCP: Kubernetes for Fabric

6. Firebase/PostgreSQL/IPFS for storage

7. **Feedback Loop**

8. Conduct user interviews

9. Log system metrics, AI accuracy, UI UX issues

📊Phase 5: Scaling & Expansion (Post Pilot)

Goals:

- Scale to state or national level
- Get policy and government adoption

Key Tasks:

1. **System Enhancement**

2. Retrain AI models on live data

3. Support IVR, regional languages, offline sync

4. **Policy Advocacy**

5. Prepare whitepaper for NITI Aayog, MeitY

6. Suggest DBT, eShram, UPI integrations

7. **India-Scale Plan**

8. Integrate India Stack: Aadhaar, DigiLocker, UPI

9. Plan phased state rollouts with local partners

📄Final Deliverables by Phase

Phase	Deliverable
1	Working PoC with wage logging chaincode + UI
2	MVP system with Fabric, ML engine, dashboards
3	Fully integrated and secured production system
4	Live pilot launch + usage feedback report
5	Scalable version + policy whitepaper

Let me know when you're ready for the Go chaincode + backend code templates.