**📌 TrustChain Full Workflow + Architecture**

**🔁 PROJECT WORKFLOW OVERVIEW**

1. **User Roles (RBAC)**:
   * **Admin**: Registers colleges and verifiers.
   * **College**: Issues certificates to students.
   * **Verifier**: Verifies certificates issued.
   * **Student**: Views and shares certificates.
2. **Main Components**:
   * **Frontend** (React)
   * **Smart Contract** (Solidity - Ethereum, Hardhat)
   * **Backend/API** (Node.js + Express)
   * **Firebase** (for Authentication + Firestore storage + Google Analytics + Hosting)
   * **Google Tools** (OAuth 2.0, Analytics, Firebase Admin SDK)

**🔐 BLOCKCHAIN STRUCTURE + RBAC REFACTORED**

Roles:

- DEFAULT\_ADMIN\_ROLE: Deployer/Admin

- COLLEGE\_ROLE: College Account

- VERIFIER\_ROLE: Verifier Account

Key Functions:

- \_setupRole(): Set during constructor

- issueCertificate(): Only COLLEGE\_ROLE

- verifyCertificate(): Only VERIFIER\_ROLE

- getCertificate(): Public read

Storage:

- mapping(bytes32 => Certificate)

- event CertificateIssued

- event CertificateVerified

**⚛️ FRONTEND ROUTING (React + Firebase Auth)**

Routes:

/ → Landing Page (info about TrustChain)

/login → Firebase Auth (Google OAuth login)

/register → Role selection (College/Verifier/Student)

/dashboard → Redirects based on user role

/dashboard/admin → Admin panel (manage colleges/verifiers)

/dashboard/college → College panel (issue certificates)

/dashboard/verifier → Verifier panel (verify certs)

/dashboard/student → Student panel (view/share certs)

/certificate/:id → Public view of certificate (IPFS/Blockchain data)

**🚀 BACKEND ROUTING (Node.js + Express)**

/api/auth/google → Google OAuth token exchange

/api/user/init → Initialize Firebase user metadata

/api/certificate/issue → POST - College issues certificate (calls smart contract)

/api/certificate/verify → POST - Verifier verifies certificate

/api/certificate/:id → GET - Get cert metadata (chain + Firestore)

/api/admin/register → POST - Admin registers college/verifier addresses

**🔥 FIREBASE IMPLEMENTATION**

* **Firebase Authentication**:
  + Google Sign-in with Firebase Auth UI
  + Role-based redirection post login
* **Firebase Firestore**:
  + Stores user metadata (role, wallet, name, email)
  + Stores off-chain cert metadata (if needed)
* **Firebase Hosting**:
  + Deploy React frontend via firebase deploy
* **Firebase Admin SDK (Node.js backend)**:
  + Validate Firebase Auth tokens
  + Role mapping and user initialization

**🧰 GOOGLE TOOLS IMPLEMENTATION**

* **Google OAuth 2.0**:
  + Via Firebase Auth for frontend login
* **Google Analytics**:
  + Track user visits, role-based flows, certificate views
  + Integrated with Firebase project
* **Google Cloud Console**:
  + Manage Firebase project, credentials, API usage

**✅ WORKFLOW IN SIMPLE STEPS**

1. Admin logs in → Registers colleges & verifiers
2. College logs in → Issues cert via form → Smart contract call → Event emitted
3. Student logs in → Views certs
4. Verifier logs in → Verifies certs
5. Public can view cert via /certificate/:id