

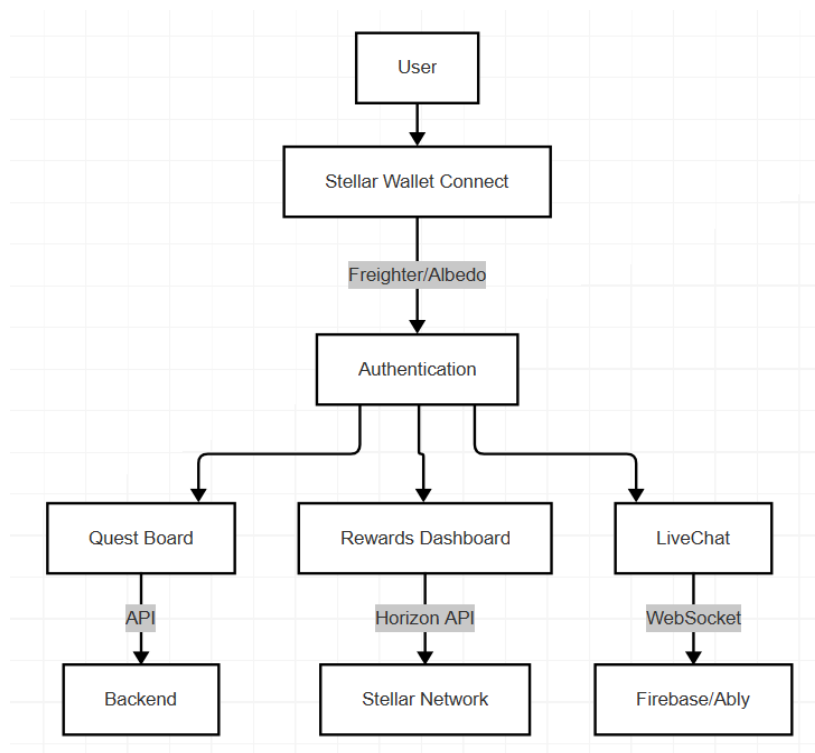
Stellearning Technical Structure

(Frontend, Backend, Blockchain, Infrastructure, and Flows)

1. Frontend (Next.js + React + Tailwind CSS)

Technologies:

- Framework: Next.js (SSR/SSG for SEO + lightweight API routes).
- UI: React + Tailwind CSS (reusable components and responsive design).
- Stellar Integration:
 - @stellar/stellar-sdk: To interact with the Stellar network.
 - @stellar/freighter-api: Login via Freighter Wallet (similar to MetaMask).
 - albedo-link: Login without extension (browser only).
- Main Components:
 - Stellar Wallet Connect: Login button with Freighter/Albedo.
 - Quest Board: List of gamified missions (rewards in tokens and badges).
 - LiveChat: Global chat (real-time).
 - Rewards Dashboard: Displays rewards and transaction history.

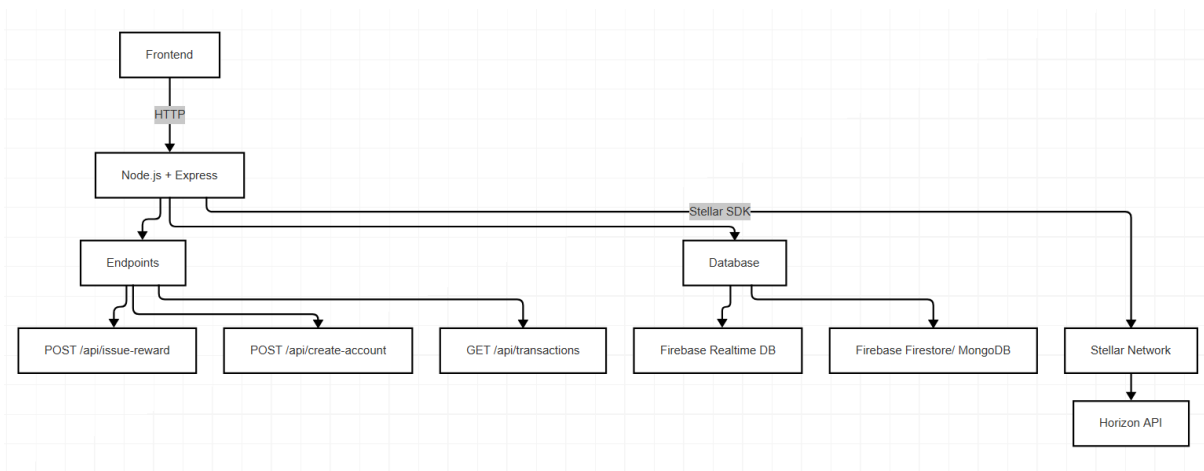


2. Backend (Node.js + Express + Firebase/MongoDB)

Main Functions:

- Manage Identity:
 - Allow new users to connect their Stellar wallets to our platform.

- Associate the user's public key with their profile in Firebase/MongoDB.
- Issue Rewards:
 - Send tokens via Stellar SDK when the user completes a mission.
- On-Chain Certification:
 - Register course completions as transactions with memo on Stellar.
- Integration with Anchors:
 - Connect to anchors like UltraStellar for deposits/withdrawals in stablecoins.
- Endpoints (API):
 - POST /api/create-account: Create a Stellar account for the user.
 - POST /api/issue-reward: Send rewards to a Stellar account.
 - GET /api/transactions: Returns the user's transaction history (via Horizon API).
- Database:
 - Firebase (MVP):
 - Firestore: User progress, missions, inventory.
 - Realtime Database: Global chat and ranking.
 - MongoDB (for scaling in Phase 2):
 - Store complex data (e.g., relationships between on-chain certificates and courses).



3. Blockchain (Stellar Network)

Configurations:

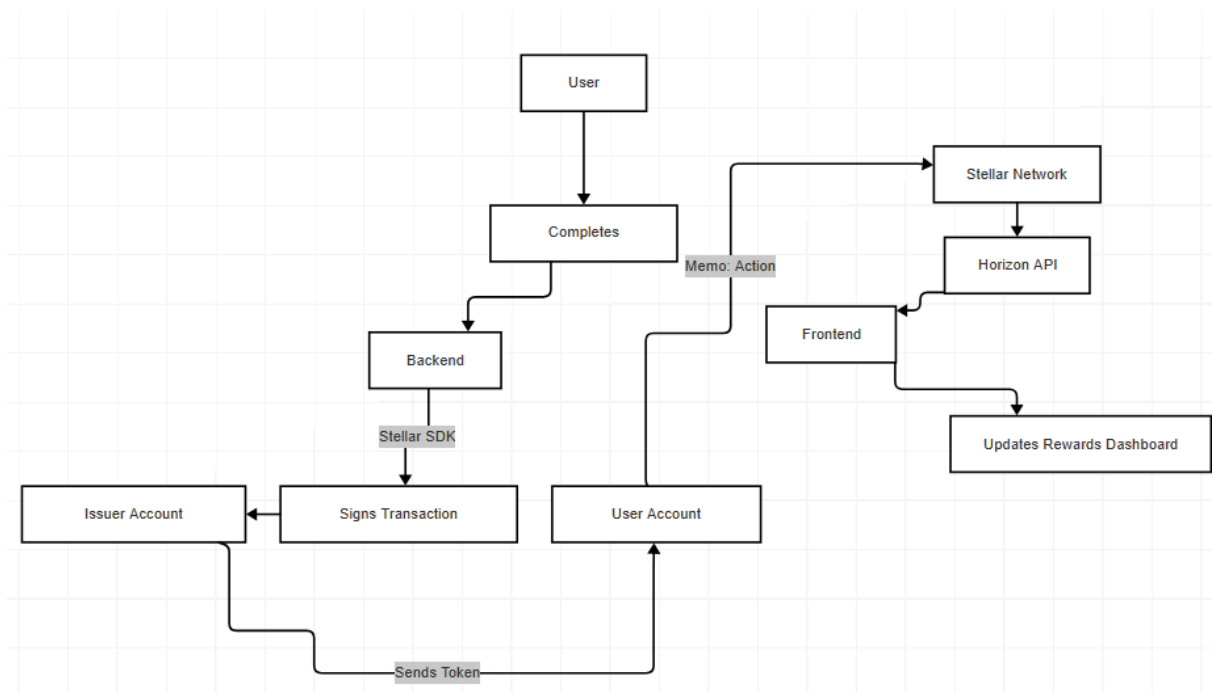
- Token:
 - Asset Type: Native Asset (custom currency).
 - Issuer: Project's master account (private key offline).
 - Supply Control: Manual issuance (managed by the backend).
- Transactions:
 - Memos: Used to link app actions to transactions.
- Rewards Flow:
 - User completes a mission in the app.
 - Backend signs a transaction sending the reward to the user's account.

- Transaction includes a memo describing the action.
- Frontend updates the user's rewards via Horizon API.

4. Gamification and Incentives

Mechanics:

- Ranking: Based on accumulated tokens (queried via Horizon API).
- NFT-like Badges (certification):
 - Represented as transactions with specific memo (e.g., `badge:web3_expert`).
 - Displayed on the profile as "collectibles."
- Missions (Quests):
 - Backend validates completion and issues rewards.
- Inventory:
 - On-Chain Items:
 - Represented by transactions with memo (e.g., avatar, tokens, certificates).
 - Off-Chain Items:
 - Stored in Firestore (e.g., profile colors, temporary accessories).



5. Infrastructure and DevOps

Hosting:

- Frontend: Vercel (automatic deployment of Next.js).

- Backend: AWS EC2 or Google Cloud Run (Node.js + Express).
- Stellar Network:
 - Horizon: Public instance (<https://horizon.stellar.org>) or self-hosted.

Monitoring:

- Sentry: For errors in frontend/backend.
- Stellar Expert: Real-time transaction explorer (e.g., <https://stellarexpert.io>).

Security:

- Private Keys:
 - Token issuer in cold storage (offline).
 - Multi-signature for critical accounts (e.g., 3/5 signatures required).

Audit:

- Transaction verification with tools like Stellar Laboratory.

