🛠️ STELLAR TOKEN PLATFORM: TECHNICAL DEVELOPMENT PRD

---

1. TECHNICAL ARCHITECTURE & STACK

This project uses the **MERN** stack combined with the Stellar ecosystem tools for a robust, scalable, and low-latency blockchain application.

Component	Technology	Rationale
Frontend	React 18, TailwindCSS, Vite	Fast development, component reusability, and modern UI implementation.
Backend	Node.js + Express, TypeScript	Real-time APIs and optimal compatibility with the Stellar SDK.
Database	MongoDB + Mongoose	Flexible schemas for handling user profiles and the complex **Binary Tree** referral structure.
Blockchain	Stellar (Soroban, Horizon API), Freighter Wallet	Enables low transaction fees (0.00001 XLM) and supports USDT stable coin for predictable returns.
Real-Time	Socket.io	Used for instant notifications (push) and live updates to the referral tree visualization.
Deployment	Render (backend), Netlify (frontend)	Cost-effective deployment solution.

2. CORE FEATURES & IMPLEMENTATION (3-Month Timeline)

The following table outlines the technical features, their implementation details, and the updated **12-Week Development Schedule**.

Feature	Description	Tech Implementation	Target Month
Wallet Connect	Secure integration with Stellar-compatible wallets (Freighter).	Stellar.js SDK, React Context, secure session management.	**Month 1 (Weeks 1-4)**
Purchase & Lock	Enable USDT→Token swap based on admin plans; enforce lock-in period.	**Soroban Smart Contract**, Stellar Horizon API, timestamp logic (Node.js/Mongoose).	**Month 1 (Weeks 1-4)**
Admin Plan Mgmt (CRUD)	Admin panel for creating, reading, updating, and deleting investment plans.	MongoDB, Express API, Role-Based Access Control (RBAC) implementation.	**Month 1 (Weeks 1-4)**
Sell & Withdraw	Post-lock, automatically swap Token→USDT and queue the withdrawal transaction.	Stellar Transaction signing, Express backend queue (non-blocking).	**Month 2 (Weeks 5-8)**
Binary Referral Tree	Visual representation of user's left/right sub-trees for up to 3 levels.		

| MongoDB nested tree structure or optimized graph representation. | Month 2 (Weeks 5-8) |

| Referral Bonuses Engine | Auto-calculate leveling bonuses (e.g., 10% L1, 5% L2). | Dedicated Cron job for scheduled bonus calculations and queuing payouts. | Month 2 (Weeks 5-8) |

| Audit & Transaction Logs | Comprehensive logs for user purchases, sells, withdrawals, and all admin actions. | MongoDB query, React table UI, Winston logger, separate audit collection. | Month 3 (Weeks 9-12) |

| Notifications | Real-time alerts for purchase, lock expiry (7 days prior), and referral bonus payouts. | SendGrid/AWS SES (Email), Socket.io, Firebase Cloud Messaging (FCM). | Month 3 (Weeks 9-12) |

---

3. API ENDPOINTS (Node.js/Express)

The backend will expose the following structured RESTful API endpoints:

Category	Endpoint	Action	Access Control
Authentication	POST /api/auth/signup	Register user, implement JWT and email verification.	Public
Plans	GET /api/plans	Fetch all active plans details.	Public
Transactions	POST /api/transactions/buy	Initiate and sign the USDT→Token purchase transaction.	Authenticated User
Transactions	POST /api/transactions/sell	Initiate Token→USDT	Authenticated User

		swap post-lock-in.	
Referrals	GET /api/referrals/tree	Fetch user's binary tree data and visualization logs.	Authenticated User
Admin	PATCH /api/admin/plans/:id	Update plan parameters (lock days, hike %).	Admin Only (RBAC)
Admin	GET /api/admin/audit-logs	Fetch all platform transactions and admin activity.	Admin Only (RBAC)

---

4. SECURITY & COMPLIANCE IMPLEMENTATION

Consideration	Implementation Details
Wallet Security	Use Freighter's secure handling; **never store private keys** on the backend; store only public addresses.
Transaction Security	Signature verification for all transactions; implement **rate limiting** (max 5 txs per minute per user).
Admin Access Control (RBAC)	Implement three roles (User, Moderator, Admin); enforce **2FA for admin accounts**; log all admin actions.

Data Protection	Encrypt sensitive fields (wallet addresses, emails); use **HTTPS + TLS 1.2+**; implement CORS security headers.
Application Security	Input validation on all endpoints; **XSS/CSRF protection**; regular dependency updates (npm audit weekly).
KYC/AML (Phase 1)	Implement email verification; ensure readiness for full KYC integration (Onfido/Sumsub) in Phase 2.

---

5. DEVELOPMENT ROADMAP (3 Months / 12 Weeks)

Month 1: Foundation & Core Transactions (Weeks 1-4)

- **Week 1 (Setup)**: Git, environment configuration, DB schema design, **Stellar SDK integration**, React boilerplate setup.
- **Week 2-3 (Core APIs)**: Authentication (JWT), Wallet Connect, **Purchase/Lock APIs**, Dashboard UI, Transaction History, Admin Plan Management CRUD.
- **Week 4 (Integration)**: Final API integration testing, implementation of the **Lock Timer**, full bug fixes and optimization for core flows.

Month 2: Referral System & Sell Flow (Weeks 5-8)

- **Week 5-6 (Referrals)**: Development of the **Binary Tree data structure**, Referral link generation, Tree visualization UI.
- **Week 7 (Monetary)**: **Sell & Withdraw API** implementation, Token→USDT swap logic, **Bonus Calculation Engine** (Cron job setup).
- **Week 8 (Polish)**: Referral activity logs, final integration testing for sell flow, and performance optimization.

Month 3: Polish, Audits, & Launch Prep (Weeks 9-12)

- **Week 9-10 (Compliance & Logs)**: Implementation of **Audit Logs** (all admin/tx actions), setup of **Email Notifications** (SendGrid) and **Push Notifications** (FCM/Socket.io).
- **Week 11-12 (Deployment)**: Complete final bug fixes, deploy to staging (Render/Netlify), run final load tests, **Production Deployment**.

6. SUCCESS METRICS & KPIs (Technical Focus)

Metric	Target	Measurement	Owner
Tx Success Rate	95%+	Confirmed txs / Initiated txs (from DB).	Backend
Avg Tx Time	<2 min	Timestamp diff (creation → Stellar confirmation).	Backend
Platform Uptime	99.9%	(Total time - downtime) / Total time * 100.	DevOps
Code Quality	80%+	Test coverage via Jest/Mocha.	QA
API Latency (p95)	<500ms	Monitoring dashboard (Datadog).	Backend
Database Query Time	<100ms	MongoDB profiling.	Backend
Error Rate	<0.5%	Sentry error tracking.	Backend