

Streple MVP – Lead Developer Technical Direction Document

Version: 1.1

Date: June 4, 2025

Prepared by: Founder's Office

For: Lead Developer & Engineering Team

MVP Scope Summary

Streple's MVP focuses on crypto-based social trading, allowing followers to automatically join trades shared by pro users, but with no real fund execution or trading in this phase.

Core MVP Features – Technical Breakdown

1. User Account System

- **Two roles:**
 - **Pro Trader** – shares trade signals
 - **Follower** – receives and reacts to signals
- **Profile data:**
 - Bio, image, stats
 - Follower count, performance history

2. Demo Balance System

- Each new user can request a \$500 demo fund.
- Balance is stored in-app (mock backend).
- Used for simulated trading only.
- Users can top up once per X period (e.g. weekly reset or via request).

3. Trade Signal Engine

- **Pro trader UI** to enter trade signal:
 - Asset (e.g. BTC/USDT)
 - Entry price (manually input or simulated)
 - Position size (% of balance)
 - Stop Loss / Take Profit (optional)
 - Exit command
- **Backend task:**
 - Push signal to all followers

- Log entry details
- Store timestamp

4. Auto-Join & Auto-Exit Simulation

- Followers who have **enabled auto-join**:
 - Trade gets copied on demo balance
 - Deduct balance virtually
 - Store open position
- When the trader **exits**:
 - Close the simulated trade
 - Update profit/loss on follower's balance
 - Notify user with result

5. Frontend Integration

- Live Feed of trades on dashboard
- Trader performance leaderboard (demo stats)
- Trade signal notification system
- Toggle for:
 - Auto-join / auto-exit
 - Manual trade simulation

Security & Compliance

- No live fund custody.
- No real crypto integrations.
- No trade execution APIs.
- **All trades and balances are simulated.**
- GDPR and EU VASP-compliant architecture (signals-only).

Technical Stack Recommendation

You may refine based on dev preferences, but initial stack suggestions:

- **Frontend**: React.js / Next.js
- **Backend**: Node.js / Nest.js
- **DB**: PostgreSQL (or Firebase Firestore)
- **Notification Engine**: Websockets or Pub/Sub (real-time update)
- **Authentication**: Google auth, normal jwt or OAuth

Deliverables & Milestones

Deliverable	Description	ETA
✓ User Role Logic	Trader vs. Follower system	Week 1

✓ Demo Balance Logic	Top-up, deduction, reset	Week 2
✓ Signal Engine	Push trades from pro to followers	Week 3
✓ Simulated Trade Module	Auto-join/exit mock trades	Week 4
✓ UI/UX Integration	Feed, profile, toggle system	Week 5
✓ Testing & Staging	Manual QA, load test	Week 6

Final Notes

- **No live crypto or fiat flows required during MVP.**
- Users feel they're "trading," but it's fully simulated.
- This architecture should be **built to easily integrate** real execution engines post-MVP with minimal refactor.