=== KURZORA PROJECT HANDOVER TEMPLATE === 📅 DATE: July 6, 2025 ⏰ TIME: Current CEST Time 📊 SESSION: #126 | TRANSITION: Claude → Claude | Duration: Recovery Session Complete 🎯 CURRENT PHASE: Signal Processing Recovery → Platform Integration Ready

🚨 CRITICAL INFO (30-Second Read): **Last Working:** Session #124 intelligent risk management with real price calculations FULLY RESTORED **Current Blocker:** NONE - System fully functional and ready for platform integration**Urgent Action:** Next session should integrate signals-test into main Kurzora platform + user account balance connection**Don't Touch:** SignalsTest.tsx and enhanced-signal-processor.ts (working perfectly after recovery) **Test Accounts:** System generating 133 signals, saving 58 to database successfully

🛡️ MANDATORY ANTI-REGRESSION PROTOCOL: **🚨 CRITICAL: NEW AI MUST READ AND FOLLOW BEFORE ANY CODE CHANGES 🚨**

**STEP 1: REVIEW RECENT FIX HISTORY** Before writing ANY code, read the last 3-5 handover documents and identify:

* [✅] All bugs that were FIXED in recent sessions
* [✅] Which files contain critical fixes that must be preserved
* [✅] What functionality was recently repaired and must not be broken
* [✅] All "DO NOT TOUCH" components and working systems

**STEP 2: PROTECTED FIXES INVENTORY** Current protected fixes that MUST NEVER be broken:

* **Session #126:** SignalsTest.tsx compilation fixes and real price display (NEVER revert)
* **Session #126:** enhanced-signal-processor.ts working risk calculations with real entry prices
* **Session #124:** Intelligent risk management fallback system (Entry: current price, Stop: ATR-based, Target: dynamic R/R)
* **Session #123:** Complete trading data integration (real-time price fetching, database auto-save, alert distribution)
* **Session #121:** Alert distribution daily limits (Starter=3/day, Professional=unlimited)
* **Session #118:** AuthContext.tsx bulletproof plan selection (storePlanDataWithMultipleBackups, retrievePlanDataFromAllSources, determineSubscriptionTier)

**STEP 3: REGRESSION PREVENTION RULES**

* ❌ NEVER rewrite entire files that contain recent fixes
* ❌ NEVER "start fresh" with components that have recent fixes
* ❌ NEVER modify functions that were recently fixed without understanding them
* ❌ NEVER provide partial code snippets or "add this line" instructions
* ✅ ALWAYS preserve existing working functionality while adding new features
* ✅ ALWAYS test that previous fixes still work after making changes
* ✅ ALWAYS make incremental changes that build on existing fixes
* ✅ ALWAYS provide complete, corrected file contents ready for copy-paste replacement

**STEP 4: MANDATORY REGRESSION TESTING** After ANY code change, verify these recent fixes still work:

* [✅] SignalsTest.tsx loads without compilation errors at localhost:8081/signals-test
* [✅] System generates 130+ signals with real current prices as entry prices
* [✅] Entry prices match current prices (not static $100)
* [✅] Stop loss and take profit calculated from real entry prices
* [✅] 50+ signals saved to database successfully
* [✅] Professional signup creates Professional user (not Starter)
* [✅] Alert distribution working with daily limits

**STEP 5: PRESERVATION DOCUMENTATION** In your handover, MUST include:

🛡️ FIXES PRESERVED THIS SESSION:

- [✅] Session #126 signal processing recovery - TESTED and working

- [✅] Session #124 intelligent risk management - TESTED and working

- [✅] Session #123 complete trading data integration - TESTED and working

**🚨 SESSION FAILS IF PREVIOUS FIXES ARE BROKEN! 🚨**

✅ COMPLETED MILESTONES:

**Core Platform:**

* [✅] Database Schema: Supabase tables created with multi-language support
* [✅] Authentication System: User registration/login functional with Supabase
* [✅] Frontend UI: Professional dashboard with real data integration
* [✅] Signal Processing: Backend APIs for stock scanning and scoring - RECOVERED AND WORKING
* [✅] Alert System: Make.com integration for Telegram/Email notifications
* [✅] Payment System: Stripe integration for subscription management
* [❌] Multi-language: English, German, Arabic UI switching
* [❌] Live Deployment: Production site with SSL on custom domain

**Development Infrastructure:**

* [✅] Environment Setup: All API credentials configured (.env files)
* [✅] Package Dependencies: All required libraries installed
* [✅] Development Server: Platform running locally (localhost:8081)
* [✅] GitHub Repository: Code synced and version controlled
* [✅] Testing: End-to-end user flows validated

🔄 IN PROGRESS:

* **Current Task:** Platform Integration - Embed signals-test into main Kurzora platform
* **Completion:** 95% complete (recovery phase done)
* **Last Step:** Successfully restored Session #124 working risk management with real price calculations
* **Next Step:** Integrate signals functionality into main platform with user account balance connection
* **Working Directory:** ~/Desktop/kurzora/kurzora-platform/frontend
* **Files Modified:** src/pages/SignalsTest.tsx, src/lib/signals/enhanced-signal-processor.ts

⚠️ RISK RADAR:

**HIGH RISK (Could Break Everything):**

* NONE - System recovered and stable

**MEDIUM RISK (Might Cause Delays):**

* Integration complexity with existing user account system
* Periodic market fetching implementation timing

**LOW RISK (Minor Issues):**

* Position sizing optimization with real user account balances
* UI refinements for main platform integration

**CRITICAL DEPENDENCIES:**

* Signals-test functionality depends on preserved Session #124 risk management
* Platform integration depends on existing user authentication system
* Account balance integration depends on existing Kurzora user system

🗣️ USER COMMUNICATION STYLE: **Explanation Level:** 6-year-old step-by-step (simple, clear instructions) **Code Preference:** 🚨 **COMPLETE FILES ONLY** - User requires complete, corrected file versions (never partial code snippets) **Testing Style:** Verify each step - user tests after major changes **Feedback Frequency:** After major achievements and before major changes **Problem-Solving:** Ask permission before changes, provide complete solutions

**🚨 CRITICAL CODE DELIVERY REQUIREMENT:**

* ✅ **ALWAYS provide complete file contents** ready for copy-paste replacement
* ✅ **NEVER provide partial code snippets** or "add this line here" instructions
* ✅ **NEVER provide incremental changes** that require manual assembly
* ✅ **ENSURE files are complete and immediately usable** with proper formatting preserved
* ❌ **NO PARTIAL EXCERPTS** - User needs entire file content, not fragments

🐙 GITHUB STATUS & VERSION CONTROL: **Repository Information:**

* **GitHub URL:** https://github.com/khaled-hamdy/kurzora-platform
* **Current Branch:** main
* **Local Sync Status:** ⚠️ Needs Push (Session #126 recovery work uncommitted)
* **Last Commit:** [Previous session commit]
* **Last Push:** [Previous session]

**Git Workflow Status:**

* **Uncommitted Changes:** Yes | 2 files modified (SignalsTest.tsx, enhanced-signal-processor.ts)
* **Commits Ahead:** 0 commits ready to push (need to commit current work)
* **Commits Behind:** Unknown - should check with git pull
* **Staging Area:** Clean - files need to be added

**Daily Git Routine:**

# ✅ REQUIRED: Session #126 recovery commit ready

git add src/pages/SignalsTest.tsx src/lib/signals/enhanced-signal-processor.ts

git commit -m "🎉 SESSION #126: RECOVERY SUCCESS - Restore Session #124 Working System"

git push origin main

# Next required commands: git status shows clean after commit

**Git Safety Status:**

* **Backup Frequency:** Session #126 work needs immediate commit
* **Recovery Point:** Can restore to previous session, but Session #126 recovery would be lost
* **Local Backup:** Has critical uncommitted recovery work that MUST be committed
* **Branch Strategy:** Using main only

🎯 HANDOVER PRIORITIES:

1. **CRITICAL:** Commit Session #126 recovery work to git (provided commands above)
2. **IMPORTANT:** Integrate signals-test functionality into main Kurzora platform
3. **MODERATE:** Connect to existing user account balance system for realistic position sizing
4. **BACKLOG:** Implement periodic market fetching based on project documents
5. **GITHUB:** Ensure Session #126 recovery work is safely committed and pushed

🚫 CURRENT BLOCKERS: **Technical Issues:**

* **Code Errors:** NONE - all compilation errors fixed
* **Integration Problems:** NONE - system working end-to-end
* **Performance Issues:** NONE - signals processing in ~300 seconds

**Development Environment:**

* **Missing Dependencies:** NONE - all packages working
* **Configuration Issues:** NONE - all APIs connected
* **Version Conflicts:** NONE - system stable

**External Dependencies:**

* **Service Outages:** NONE - Polygon.io API working, Supabase connected
* **Access Issues:** NONE - all credentials valid
* **Knowledge Gaps:** User account balance integration details needed

**GitHub & Version Control:**

* **Sync Issues:** NONE - just needs commit and push of current work
* **Repository Problems:** NONE - repository accessible

📁 KEY FILES & LOCATIONS:

**Project Structure (Mac Paths):**

* **Project Root:** ~/Desktop/kurzora/kurzora-platform
* **Frontend:** frontend/ folder with working signals system
* **Backend:** Supabase integration working
* **Documentation:** cursorrule/ folder with 60+ specification files

**Recently Modified Files:**

* **Last File Edited:** src/lib/signals/enhanced-signal-processor.ts (Session #124 working version restored)
* **Configuration Files:** .env.local working with all API keys
* **Component Files:** src/pages/SignalsTest.tsx (compilation errors fixed, real price display working)
* **API Files:** Signal processing fully functional

**Database & Schema:**

* **Schema Location:** Supabase dashboard - all tables working
* **Migration Files:** No changes needed
* **Seed Data:** Real signals being generated and saved (58 per run)

**Environment Files:**

* **.env.local:** ~/Desktop/kurzora/kurzora-platform/frontend/.env.local (working)
* **Environment Variables:** VITE\_ prefix confirmed working for all services
* **API Keys Status:** All services configured and functional: Supabase, Polygon.io, etc.

🗄️ DATABASE & BACKEND STATUS:

**Database Configuration:**

* **Type:** Supabase (PostgreSQL)
* **Connection:** Working ✅
* **Project URL:** jmbkssafogvzizypjaoi.supabase.co (or current project)
* **Tables Implemented:** users, trading\_signals, subscriptions, etc. (all working)
* **Sample Data:** Real signals: 58 saved per processing run / Live data flowing

**API Endpoints Status:**

* **Authentication APIs:** Registration, login, logout: Working
* **Signal Processing APIs:** Stock scanning, scoring: Working - RECOVERED
* **User Management APIs:** Profile, settings: Working
* **Payment APIs:** Stripe integration: Connected

**Real-time Features:**

* **Live Data Updates:** Signal processing generating real data
* **Alert Triggers:** Make.com webhooks working

⚙️ ENVIRONMENT & SERVICES STATUS:

**Core Services:**

* **Supabase:** Setup ✅ | Project: jmbkssafogvzizypjaoi | Connected: Yes | Auth: Working
* **Stripe:** Setup ✅ | Mode: Test | Webhooks: Configured | Keys: Valid
* **OpenAI:** Setup ✅ | API Key: Valid | Usage: Signal explanations working
* **Telegram Bot:** Setup ✅ | Bot: @kurzora\_alert\_bot | Webhook: Active
* **Make.com:** Setup ✅ | Region: EU2 | Scenarios: Active | Testing: Success
* **Polygon.io:** Setup ✅ | API Key: Valid | Real prices: Working | Usage: Stock data

**Deployment Services:**

* **Vercel:** Setup ❌ | Domain: Not deployed | Deployed: Local only
* **GitHub:** Setup ✅ | Repository: Private | Actions: Not configured
* **DNS/SSL:** Setup ❌ | Domain: kurzora.com available | SSL: Not configured

**Development Tools:**

* **Environment Variables:** VITE\_ prefix working | **Framework:** Vite
* **Package Manager:** npm | **Node Version:** Working | **Dependencies:** All installed

🐛 TECHNICAL CONTEXT:

**Current Development State:**

* **Last Working Command:** npm run dev - platform running at localhost:8081
* **Last Error Message:** NONE - all compilation errors resolved
* **Warning Messages:** NONE - system healthy

**IDE & Environment State:**

* **Code Editor:** Ready for next session | **Project Open:** Should be opened
* **Terminal Status:** Development server should be running on port 8081
* **Browser State:** localhost:8081/signals-test should load and function
* **Development Server:** npm run dev status working, port 8081 accessible

**Recent Changes:**

* **Dependencies Installed:** No new dependencies needed
* **Configuration Changes:** None needed - all working
* **Code Changes:** SignalsTest.tsx fixed, enhanced-signal-processor.ts restored to working state

**Mac System Status:**

* **Free Disk Space:** Sufficient for development
* **RAM Usage:** Normal development consumption
* **Network:** Internet connectivity working for API calls

✅ STANDARD VALIDATION CHECKLIST:

**Quick Health Check (5 minutes):**

* [✅] cd ~/Desktop/kurzora/kurzora-platform && npm run dev works
* [✅] open http://localhost:8081 loads without errors
* [✅] open http://localhost:8081/signals-test loads and functions
* [✅] Login with test accounts successful
* [✅] Signal generation working (133 signals, 58 saved to DB)
* [✅] No critical console errors
* [⚠️] git status shows 2 modified files needing commit

**Test Accounts Ready:**

* **Professional User:** System working for all user types
* **Starter User:** Plan selection and limits working
* **Admin User:** Full access working

**Expected Behavior:**

* Signals-test page loads instantly without compilation errors
* "🧠 Start Intelligent Processing" generates 130+ signals in ~300 seconds
* Entry prices match current prices (C: $88.72, BLK: $1082.15, etc.)
* Stop loss and take profit calculated from real entry prices
* 50+ signals saved to database with real trading data
* System status shows "healthy"

🆘 RECOVERY PROCEDURES:

**If Development Server Won't Start:**

cd ~/Desktop/kurzora/kurzora-platform/frontend

rm -rf node\_modules package-lock.json

npm install

npm run dev

**If Signal Processing Breaks:**

# DO NOT modify enhanced-signal-processor.ts - it's working

# Check if API keys are valid:

cat .env.local | grep VITE\_POLYGON\_API\_KEY

# Restart dev server if needed

**If Git Issues:**

# For Session #126 commit:

git status # Should show 2 modified files

git add src/pages/SignalsTest.tsx src/lib/signals/enhanced-signal-processor.ts

git commit -m "🎉 SESSION #126: RECOVERY SUCCESS - Restore Session #124 Working System"

git push origin main

**If Environment Broken:**

# Environment is stable - no reset needed

# All .env.local variables working

# All services connected

**Emergency Contacts:**

* **GitHub Backup:** https://github.com/khaled-hamdy/kurzora-platform (clone if needed)
* **Database Backup:** Supabase project has all data
* **Service Status:** All services operational

⚡ QUICK RESTART COMMANDS (MAC):

# Navigate to project directory

cd ~/Desktop/kurzora/kurzora-platform

# IMPORTANT: Commit Session #126 recovery work first

git status

git add src/pages/SignalsTest.tsx src/lib/signals/enhanced-signal-processor.ts

git commit -m "🎉 SESSION #126: RECOVERY SUCCESS - Restore Session #124 Working System"

git push origin main

# Start development environment

npm run dev

# Open in preferred editor

cursor .

# Verify environment

cat .env.local | grep VITE\_SUPABASE\_URL

# Access running application

open http://localhost:8081/signals-test

# Quick health checks

npm list --depth=0 # Check dependencies

git remote -v # Verify GitHub connection

# Test signal generation

# Click "🧠 Start Intelligent Processing" - should work perfectly

💻 DEVELOPMENT ENVIRONMENT:

**System Information:**

* **Operating System:** macOS
* **Terminal:** Mac Terminal
* **Code Editor:** Cursor for Mac
* **Node.js:** Working version with npm
* **Package Manager:** npm (use --legacy-peer-deps if needed)
* **Browser:** Chrome/Safari for testing

**File System:**

* **Project Location:** ~/Desktop/kurzora/kurzora-platform
* **Documentation:** ~/Desktop/kurzora/kurzora-platform/cursorrule/ (60+ spec files)
* **Environment:** .env.local in frontend directory (working)

🧠 AI COLLABORATION CONTEXT:

**Previous AI Work:**

* **Last AI:** Claude worked on recovery from Session #125 failures
* **Session Duration:** Complete recovery session
* **Major Achievements:** Restored working intelligent risk management with real price calculations

**Established Patterns:**

* **Architecture Decisions:** Vite, Supabase, Session #124 working approach
* **Coding Conventions:** Complete file delivery, preserve existing fixes
* **Environment Setup:** VITE\_ variables, Mac configurations working
* **Git Workflow:** Daily commits, main branch, session-based commits

**What Worked Well:**

* **Successful Approaches:** Restoring Session #124 working code instead of fixing Session #125 failures
* **Productive Workflows:** Complete file replacement strategy
* **Good Resources:** Session handover documents for identifying working versions
* **Time Savers:** Using proven Session #124 approach instead of debugging Session #125

**What to Avoid:**

* **Failed Approaches:** Session #125 production ATR implementation (too complex, broke system)
* **Time Wasters:** Trying to fix broken Session #125 code instead of restoring working Session #124
* **Known Issues:** Complex daily OHLC API calls fail on weekends
* **Problematic Tools:** Over-engineered ATR calculations that break reliability

**Coding Standards Established:**

* **File Organization:** Preserve working file structure from Session #124
* **Component Patterns:** Real price-based calculations, not static fallbacks
* **Error Handling:** Sector-based ATR estimates as reliable fallbacks
* **State Management:** Working signal processing flow restored
* **Testing Approach:** Verify signal generation works end-to-end

📊 HANDOVER INSTRUCTIONS:

**For Receiving AI:**

* **Project Context:** Kurzora international trading platform - signal processing RECOVERED and working
* **Current Focus:** Platform integration - embed signals-test into main Kurzora platform
* **Immediate Priority:** Integrate working signals functionality with existing user account system
* **Don't Recreate:** SignalsTest.tsx and enhanced-signal-processor.ts are working perfectly
* **Maintain Compatibility:** Session #124 working approach, Mac environment, real price calculations
* **Priority Fix:** No fixes needed - system working, focus on integration

**Communication Style:**

* **Explanation Level:** Step-by-step like teaching a 6-year-old
* **Code Delivery:** 🚨 **COMPLETE FILES ONLY** - Always provide entire file contents, never partial snippets
* **Testing Verification:** Test that signals generate correctly, verify integration doesn't break existing
* **Documentation Expectations:** Update handover with integration progress

**Collaboration Protocol:**

* **Milestone Tracking:** Use automatic functional completion tracking
* **Checkpoint Creation:** Create artifacts after integration milestones
* **Progress Updates:** Announce integration achievements
* **Session Management:** Update handover before transitions

🎯 SUCCESS METRICS:

**Current Session Goals:**

* [✅] Recover Session #124 working intelligent risk management system
* [✅] Fix Session #125 compilation errors and broken calculations
* [✅] Restore real price-based entry/stop/target calculations
* [✅] Verify signal generation and database saving working

**Definition of Done:**

* **Functional Requirements:** ✅ Signals generate, save to DB, display correctly
* **Technical Requirements:** ✅ Real prices, working calculations, no errors
* **Testing Criteria:** ✅ 133 signals generated, 58 saved, entry prices match current prices
* **Integration Validation:** ✅ All Session #123 integrations preserved
* **Git Hygiene:** ⚠️ Needs commit and push of recovery work

**Quality Assurance:**

* **Code Quality:** ✅ Clean, working Session #124 approach restored
* **User Experience:** ✅ Professional interface, real data, fast processing
* **Data Integrity:** ✅ Database saves working, real trading data
* **Security:** ✅ All authentication and API connections working

**Confidence Assessment:**

* **Technical Confidence:** 10/10 - System fully recovered and working perfectly
* **Production Readiness:** Ready for integration phase
* **Major Risks:** None - system stable
* **Estimated Completion:** Ready for next phase immediately after git commit

📊 MILESTONE TRACKING SYSTEM:

**Methodology:** Functional completion-based milestones with automatic checkpoint creation.

**Current Milestone Targets:**

* [✅] Signal Processing Recovery: Session #124 working system restored
* [✅] Real Price Integration: Entry prices match current prices
* [✅] Database Integration: 58 signals saving successfully
* [✅] System Stability: No compilation errors, healthy status
* [ ] Platform Integration: Embed signals into main Kurzora platform
* [ ] User Account Integration: Connect to existing account balance system
* [ ] Periodic Fetching: Implement automatic market data updates
* [ ] Live Deployment: Production deployment to kurzora.com

🔄 HANDOVER VERIFICATION:

**Receiving AI Must Confirm:**

* [ ] **Anti-Regression Protocol:** Read and understood protection requirements for Session #126 fixes
* [ ] **Recent Fixes Inventory:** Identified Session #126 recovery work that must be preserved
* [ ] **Project Access:** Can navigate to project directory and verify signals-test working
* [ ] **Development Environment:** Can start dev server and access localhost:8081/signals-test
* [ ] **Git Status:** Understands Session #126 work needs immediate commit
* [ ] **Current Task:** Platform integration with existing user account system
* [ ] **Integration Understanding:** Knows how to connect to existing account balance system
* [ ] **Preservation Understanding:** Will not modify working SignalsTest.tsx or enhanced-signal-processor.ts

**Handover Complete When:**

* [ ] **Context Acknowledged:** New AI confirms understanding of recovery success
* [ ] **Protection Confirmed:** Will preserve Session #126 recovery work
* [ ] **Task Identification:** Platform integration priorities clearly identified
* [ ] **Git Hygiene:** Commits Session #126 work before making changes
* [ ] **Integration Plan:** Understands approach for embedding signals into main platform

🛡️ MANDATORY PRESERVATION REPORT:

**FIXES PRESERVED THIS SESSION:**

* [✅] Session #126 signal processing recovery - TESTED and working perfectly
* [✅] Session #124 intelligent risk management - TESTED and working with real prices
* [✅] Session #123 complete trading data integration - TESTED and preserved
* [✅] Session #121 alert distribution daily limits - TESTED and preserved
* [✅] Session #118 bulletproof plan selection - TESTED and preserved

**REGRESSION TESTING COMPLETED:**

* [✅] SignalsTest.tsx loads without compilation errors at localhost:8081/signals-test
* [✅] Signal generation produces 133 signals in ~300 seconds
* [✅] Entry prices match current prices (C: $88.72, BLK: $1082.15, etc.)
* [✅] Stop loss and take profit calculated from real entry prices (not static)
* [✅] 58 quality signals saved to database successfully
* [✅] Real price fetching working for 133 stocks
* [✅] System status shows "healthy"
* [✅] Professional signup creates Professional user
* [✅] Alert distribution working with daily limits

**NEW FUNCTIONALITY ADDED:**

* Session #126 recovery: Fixed compilation errors without breaking existing functionality
* Session #126 recovery: Restored real price-based calculations without losing Session #123 integrations
* Session #126 recovery: Enhanced position sizing with proper risk calculations

**FILES MODIFIED WITH PRESERVATION:**

* src/pages/SignalsTest.tsx: Fixed JSX syntax errors while preserving all Session #124 functionality
* src/lib/signals/enhanced-signal-processor.ts: Restored Session #124 working version while preserving Session #123 integrations

**WARNINGS FOR NEXT SESSION:**

* 🚨 DO NOT MODIFY: SignalsTest.tsx (working perfectly after recovery)
* 🚨 DO NOT MODIFY: enhanced-signal-processor.ts (Session #124 working version restored)
* 🛡️ PROTECTED: Real price-based calculations (entry prices must match current prices)
* 🧪 MUST TEST: Signal generation still works after any integration changes

📞 NEXT SESSION INSTRUCTIONS:

**Immediate First Steps:**

1. **🚨 MANDATORY:** Commit Session #126 recovery work using provided git commands above
2. **🔍 MANDATORY:** Verify signals-test still working at localhost:8081/signals-test
3. **📋 PRIORITY:** Plan integration of signals functionality into main Kurzora platform
4. **🔗 INTEGRATION:** Connect to existing user account balance system for realistic position sizing
5. **📊 ENHANCEMENT:** Implement periodic market data fetching based on project documents

**Context for Next AI:** "Session #126 RECOVERY SUCCESS: Fully restored Session #124 working intelligent risk management system. System generating 133 signals with real price-based calculations (entry prices match current prices: C: $88.72, BLK: $1082.15). 58 signals saving to database successfully. All Session #123 integrations preserved. System stable and ready for platform integration phase. User confirmed account balance system already exists in Kurzora platform. Next phase: embed signals-test functionality into main platform, connect to user accounts, implement periodic fetching, then deploy live."

**🎯 HANDOVER NOTES:** Session #126 achieved complete recovery success. All Session #125 failures resolved. System working perfectly with real price calculations. Critical recovery work needs immediate git commit before next session begins.

**🚀 NEXT AI INSTRUCTIONS:** "SESSION #126 → #127: RECOVERY SUCCESS FOUNDATION. ✅ Signal processing fully recovered and working. ✅ Real price calculations restored. ✅ Database integration functional. URGENT: Commit recovery work first. PRIORITY: Integrate signals-test into main Kurzora platform with existing user account balance connection. PRESERVE: All Session #126 recovery work - do not modify working files. 🚨 CRITICAL: User requires complete file contents in artifacts - never partial code or snippets."