📚 MANDATORY STARTUP SEQUENCE **STEP 1: Search Project Knowledge**

Use project\_knowledge\_search to find and read ALL project documents:

- Latest handover document

- Database schema and specifications

- Development status and progress notes

**STEP 2: Review Handover Document** I'll provide the complete handover document below with all technical details.

🛡️ CRITICAL SAFETY PROTOCOLS **🚨 ABSOLUTE REQUIREMENTS:**

1. **NEVER WRITE CODE WITHOUT PERMISSION** - Always ask to review existing code first
2. **ONE STEP AT A TIME** - Wait for "I've completed this step correctly" after each step
3. **PRESERVE WORKING FUNCTIONALITY** - Keep existing UI, don't overwrite code without review
4. **COMPLETE CODE ONLY** - Provide full, production-ready code in artifacts

🎯 IMMEDIATE TASK & PRIORITIES **CURRENT TASK:** Complete Multi-User Telegram Connection Flow + Automatic Subscription Management

**COMPLETION STATUS:** Tiered alert system 100% complete, Telegram connection 33% complete (founder receiving Telegram alerts, need connection flow for other professional users)

**IMMEDIATE NEXT STEP:** Build Telegram bot /start command to capture chat IDs for professional users + implement subscription status checking

**FILES TO REVIEW FIRST:**

* File 1: Supabase Edge Function multi-user-alerts - review working tier logic (DON'T CHANGE)
* File 2: Telegram bot @kurzora\_alert\_bot current configuration - check existing commands
* File 3: Frontend Telegram connection components - see what UI exists for professional users

**CURRENT BLOCKER:** Professional users (khaled.hamdy.hassan@gmail.com, hhkhalid@hotmail.com) have no way to connect their Telegram accounts to receive alerts

**CRITICAL CONTEXT:** 🎉 HISTORIC BREAKTHROUGH - Session #74 achieved complete tiered alert system! Starter users (3/day limit, email only) vs Professional users (unlimited, email + Telegram) working perfectly. DON'T recreate this - enhance it with multi-user Telegram connection.

🔧 TECHNICAL CONTEXT **WHAT'S WORKING:**

* Complete tiered alert system with daily limits validated
* Edge Function multi-user-alerts with tier-based logic deployed
* Database schema with telegram\_chat\_id, daily\_alerts\_sent, tier columns
* Make.com bulletproof scenario with error handlers for Email + Telegram
* 8 users configured: 5 starter + 3 professional with tier settings
* Score filtering (65+ threshold) for both tiers working
* Founder receiving Telegram alerts via @kurzora\_alert\_bot
* Daily limit blocking for starter users confirmed working

**WHAT'S BROKEN:**

* Professional users missing telegram\_chat\_id values in database
* No Telegram bot /start command to capture chat IDs
* No frontend UI for professional users to connect Telegram accounts
* No subscription status checking (cancelled users still get alerts)
* Only founder has Telegram connection, other professional users excluded

**LAST SUCCESSFUL COMMAND:**

INSERT INTO trading\_signals (ticker, signal\_type, confidence\_score, signals, entry\_price, final\_score, status)

VALUES ('TSLA', 'bullish', 70, '{"1H": 72, "4H": 68}'::jsonb, 250.00, 70, 'active');

**CURRENT ERROR (if any):** None - system working perfectly after Session #74 breakthrough

**ENVIRONMENT STATUS:** localhost:8081 working, npm run dev operational, Edge Function deployed live, Supabase connected, tiered system validated

🎯 SESSION STARTUP WORKFLOW **REQUIRED STARTUP RESPONSE:**

"I've read your project knowledge and handover document.

Current task: Complete Multi-User Telegram Connection Flow + Automatic Subscription Management

Next step: Build Telegram bot /start command to capture chat IDs for professional users + implement subscription status checking

Before proceeding, may I review: Supabase Edge Function multi-user-alerts, Telegram bot configuration, and frontend Telegram components?

I'll work one step at a time and wait for confirmation."

**DEVELOPMENT PATTERN:**

1. "Let me review [specific file] first"

2. "Here's my analysis and proposed approach"

3. "May I proceed with this change?"

4. WAIT for "Yes"

5. Implement change

6. "Please confirm this step works"

7. WAIT for confirmation

🚨 SPECIAL INSTRUCTIONS **URGENT:** Commit Session #74 tiered alert system breakthrough to GitHub before making new changes

**DON'T TOUCH:**

* Working Edge Function multi-user-alerts (perfect tier logic)
* Working Make.com scenario with error handlers
* Working database tier columns and user configurations
* Existing tiered alert distribution system (starter 3/day vs professional unlimited)

**INVESTOR DEMO:** Not specified / Platform has working tiered business model now

**GIT SAFETY:** CRITICAL - Need immediate commit of Session #74 breakthrough before any new development

⚠️ CRITICAL WARNINGS **NEVER:**

* ❌ Write code without permission
* ❌ Change working Edge Function tier logic
* ❌ Recreate working tiered alert system
* ❌ Make multiple changes simultaneously

**ALWAYS:**

* ✅ Ask to review existing implementation first
* ✅ Wait for step-by-step confirmation
* ✅ Preserve all working tiered functionality
* ✅ Provide complete code artifacts

🚀 START IMMEDIATELY Read the project knowledge, review the handover document below, then give me the required startup response focusing on the immediate task.

**Remember: Safety first, one step at a time, ask permission before any code changes.**

**📋 HANDOVER DOCUMENT:** [PASTE THE COMPLETE SESSION #75 HANDOVER DOCUMENT HERE]

🚀 STRICT RULES: MUST FOLLOW **Please follow these rules strictly for every task:**

1. **Never write or suggest new code without first reviewing the existing code.**
2. **Do not ask me to search or modify code manually.** Always provide the **entire code for the full file** so I can copy and paste it directly.
3. **Proceed step by step.** After each step, wait for my confirmation before moving to the next.