📚 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:** Plan logic precision fix - AuthContext plan determination creating all users as Professional instead of respecting Starter vs Professional selection

**COMPLETION STATUS:** 99% complete - Subscription system fully operational, Stripe integration working, only plan selection logic needs precision tuning

**IMMEDIATE NEXT STEP:** Fine-tune determineSubscriptionTier() function in AuthContext.tsx to respect explicit plan selection while maintaining smart defaults

**FILES TO REVIEW FIRST:**

* File 1: frontend/src/contexts/AuthContext.tsx - Review determineSubscriptionTier() function (lines 58-128) for plan logic over-correction
* File 2: frontend/src/components/auth/SignupForm.tsx - Verify plan data passing correctly to signUp function

**CURRENT BLOCKER:** Plan determination logic over-corrected after long-term solution - all users created as Professional tier regardless of Starter/Professional selection

**CRITICAL CONTEXT:** SESSION #62 COMPLETE SUCCESS! Entire subscription system operational (signup → Stripe customers → database). Nuclear test proved all components work perfectly. Only refinement needed: plan selection accuracy.

🔧 TECHNICAL CONTEXT **WHAT'S WORKING:** ✅ Complete subscription system (signup → user creation → Stripe customer) ✅ Frontend forms passing plan data correctly ✅ Backend API subscription processing (localhost:3001) ✅ Frontend development server (localhost:8081) ✅ Stripe integration creating customers with correct metadata ✅ Database user creation (just wrong tier assignment) ✅ AuthContext, SignupForm, LandingPage all functional

**WHAT'S BROKEN:** ❌ Plan determination logic: All users created as Professional (including Starter selections) ❌ Stripe shows plan\_id: "starter" correctly, Supabase shows subscription\_tier: "professional" incorrectly

**LAST SUCCESSFUL COMMAND:** Both servers running: frontend (localhost:8081) + backend (localhost:3001)

**CURRENT ERROR (if any):** No errors - system working perfectly except plan tier assignment logic

**ENVIRONMENT STATUS:** localhost:8081 frontend operational, localhost:3001 backend operational, all APIs connected, subscription flow end-to-end working

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

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

Current task: Plan logic precision fix - AuthContext plan determination creating all users as Professional instead of respecting Starter vs Professional selection

Next step: Fine-tune determineSubscriptionTier() function in AuthContext.tsx to respect explicit plan selection while maintaining smart defaults

Before proceeding, may I review: frontend/src/contexts/AuthContext.tsx and frontend/src/components/auth/SignupForm.tsx?

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:** Plan logic precision fix - simple refinement needed for production-ready system

**DON'T TOUCH:** Beautiful Lovable UI (perfect), working subscription system (just completed), backend APIs (operational), pricing consistency ($19/$49), port configurations (8081/3001)

**INVESTOR DEMO:** No specific deadline mentioned - system nearly production-ready

**GIT SAFETY:** URGENT - Major achievements from Session #62 need commit before any new modifications

⚠️ CRITICAL WARNINGS **NEVER:**

* ❌ Write code without permission
* ❌ Change UI without explicit request
* ❌ Make multiple changes simultaneously
* ❌ Recreate working subscription system components

**ALWAYS:**

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

🚀 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:** === KURZORA PROJECT HANDOVER TEMPLATE === 📅 DATE: June 26, 2025 ⏰ TIME: 20:45 CEST 📊 SESSION: #62 | TRANSITION: Claude → Next Claude | Duration: 3 hours (Complete Success + 1 Bug) 🎯 CURRENT PHASE: Subscription System Fine-Tuning - Plan Logic Precision Fix

**✅ COMPLETED MILESTONES:**

**Core Platform:**

* [✅] **Database Schema**: Supabase tables operational with user/subscription structure
* [✅] **Authentication System**: User registration/login functional with tier creation
* [✅] **Frontend UI**: Professional Lovable dashboard with perfect $19/$49 pricing consistency
* [✅] **Signal Processing**: Backend APIs operational (not focused this session)
* [✅] **Alert System**: Make.com integration working (not focused this session)
* [✅] **Payment System**: 🎉 **COMPLETE** - Stripe integration fully operational with customer creation
* [❌] **Multi-language**: English working, German/Arabic not implemented
* [❌] **Live Deployment**: Currently running on localhost:8081 (frontend) + localhost:3001 (backend)

**Development Infrastructure:**

* [✅] **Environment Setup**: All API credentials configured with VITE\_ prefix working perfectly
* [✅] **Package Dependencies**: All required libraries installed for frontend + backend
* [✅] **Development Server**: Both servers operational - Frontend localhost:8081 + Backend localhost:3001
* [✅] **GitHub Repository**: Code synced with development branch
* [✅] **Testing**: Subscription system validated with nuclear test (all users → Professional)

**🔄 IN PROGRESS:**

* **Current Task:** **CRITICAL** - Plan logic precision fix (over-corrected to all Professional)
* **Completion:** 99% complete - Subscription system works, needs plan selection refinement
* **Last Step:** Implemented long-term solution with robust plan determination logic
* **Next Step:** **IMMEDIATE** - Fine-tune plan logic to respect Starter vs Professional selection
* **Working Directory:** ~/Desktop/kurzora/kurzora-platform/frontend/src/contexts/AuthContext.tsx
* **Files Modified:** AuthContext.tsx (long-term solution implemented), LandingPage.tsx (fixed), SignupForm.tsx (fixed)

**🚫 CURRENT BLOCKERS:**

**Technical Issues:**

* **Plan Logic Over-Correction**: All users created as Professional (including Starter selections)
* **Stripe-Supabase Mismatch**: Stripe correctly shows plan\_id: "starter", Supabase shows subscription\_tier: "professional"

**EXACT SOLUTION NEEDED:**

* **File:** frontend/src/contexts/AuthContext.tsx
* **Function:** determineSubscriptionTier() (lines 58-128)
* **Issue:** Logic defaults to Professional too aggressively
* **Fix:** Respect explicit plan selection while maintaining smart defaults

**WORKING SYSTEMS:**

* ✅ **Frontend:** localhost:8081 with working subscription flow
* ✅ **Backend:** localhost:3001 with operational subscription processing
* ✅ **Stripe Integration:** Creating customers with correct metadata
* ✅ **Database:** Storing users (just with wrong tier)

**📞 NEXT SESSION INSTRUCTIONS:**

**Immediate First Steps:**

1. cd ~/Desktop/kurzora/kurzora-platform
2. git add . && git commit -m "🎉 SESSION #62: SUBSCRIPTION SYSTEM COMPLETE + Long-term plan logic"
3. git push origin backend-development
4. Start both servers: backend (3001) + frontend (8081)
5. **Fine-tune plan determination logic in AuthContext.tsx**

**Context for Next AI:** "SESSION #62 COMPLETE SUCCESS: Entire subscription system operational! End-to-end signup → Stripe customer creation working perfectly. Nuclear test proved all components functional. Implemented long-term plan determination architecture. ONLY REMAINING ISSUE: Plan logic over-corrected - all users become Professional instead of respecting Starter vs Professional selection. Need precision fix in AuthContext.tsx determineSubscriptionTier() function. All systems proven working - simple logic refinement needed."

🚀 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.