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

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

- Latest handover document (Session #57 - User Signup Fixed, Subscription Processing Next)

- Database schema and specifications (Comprehensive Supabase setup)

- Development status and progress notes (57 sessions of development)

**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:** Build backend subscription processing to convert captured payment methods into active Stripe subscriptions

**COMPLETION STATUS:** 80% complete - User signup fixed and working, payment methods captured, need subscription creation backend

**IMMEDIATE NEXT STEP:** Build backend API endpoints to process stored payment methods and create actual Stripe subscriptions with 7-day trial

**FILES TO REVIEW FIRST:**

* File 1: src/components/auth/SignupForm.tsx - to understand how payment methods are currently stored in localStorage
* File 2: src/contexts/AuthContext.tsx - to see user creation flow and where subscription processing should integrate
* File 3: Database schema - to understand subscription-related table structure

**CURRENT BLOCKER:** No backend logic exists to convert payment methods into Stripe subscriptions (payment captured but no subscription created)

**CRITICAL CONTEXT:** Session #57 MAJOR SUCCESS: Fixed signup function name mismatch (signup → signUp), users can now create accounts successfully! Email confirmations work, payment methods captured. Minor redirect bug exists (users stay on signup page) but is secondary priority.

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

* Beautiful Lovable signup UI with perfect $19/$49 pricing consistency
* User account creation and authentication (JUST FIXED!)
* Supabase database with comprehensive schema
* Stripe payment method collection working perfectly
* Professional Kurzora-branded email confirmations
* Admin panel and existing user management
* Development server stable on localhost:8081

**WHAT'S BROKEN:**

* No backend subscription processing (payment methods stored in localStorage only)
* Signup redirect issue (users stay on signup page after successful account creation)
* No actual Stripe subscriptions created (trial management missing)

**LAST SUCCESSFUL COMMAND:** npm run dev (localhost:8081 working perfectly with functional signup)

**CURRENT ERROR (if any):** No blocking errors - minor Stripe console warnings but functionality unaffected

**ENVIRONMENT STATUS:** localhost:8081 working perfectly, all services operational, development server stable, signup flow functional

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

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

Current task: Build backend subscription processing to convert captured payment methods into active Stripe subscriptions

Next step: Build backend API endpoints to process stored payment methods and create actual Stripe subscriptions with 7-day trial

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

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

**DEVELOPMENT PATTERN:**

1. "Let me review SignupForm.tsx first to understand payment method storage"

2. "Here's my analysis and proposed approach for subscription backend"

3. "May I proceed with creating subscription processing functions?"

4. WAIT for "Yes"

5. Implement subscription backend

6. "Please confirm this step works"

7. WAIT for confirmation

🚨 SPECIAL INSTRUCTIONS **URGENT:** Commit Session #57 signup fixes before making any new changes (critical Git safety)

**DON'T TOUCH:**

* Beautiful Lovable UI design (perfect as-is)
* Working signup flow (just fixed successfully)
* Perfect pricing system ($19/$49 consistency achieved)
* Admin panel layout (working perfectly)
* Email confirmation system (professionally branded)

**INVESTOR DEMO:** Not specified - focus on completing subscription processing for full user journey

**GIT SAFETY:** CRITICAL - Must commit Session #57 signup fixes before starting subscription backend work

⚠️ CRITICAL WARNINGS **NEVER:**

* ❌ Write code without permission
* ❌ Change beautiful Lovable UI design
* ❌ Modify working signup flow or pricing
* ❌ Make multiple changes simultaneously

**ALWAYS:**

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

🚀 START IMMEDIATELY Read the project knowledge, review the handover document below, then give me the required startup response focusing on building subscription processing backend.

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

**📋 HANDOVER DOCUMENT:** [PASTE COMPLETE SESSION #57 HANDOVER DOCUMENT BELOW]

🚀 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.