CtrlSketch Pro Development Roadmap

# 🧱 PHASE 1: The Solid Core

Goals: Bulletproof the canvas engine — everything testable, no fluff.

Must-Haves:  
- Full Tailwind styling enabled  
- Drawing canvas (SVG or HTML5)  
- Rectangle, Line, Arrow, Label tools  
- Zoom, Pan, Undo/Redo  
- Snap-to-grid + grid toggle  
- Select + move + resize shapes  
- Properties panel (right flyout)  
- Right-click menu (delete, props)

# 📚 PHASE 2: Shape Library + Multi-Page

Goals: Build layout foundation for repeatable drawing workflows  
  
Features:  
- Shape palette with drag/drop  
- Shape metadata (type, tag, part#, qty)  
- Multi-page system (new page, nav tabs)  
- Autosave current drawing state  
- Page titles + descriptions  
- Page-level properties (e.g., drawing type)

# 🧠 PHASE 3: AI + Takeoff Tools

Goals: Layer in smarts — don’t hallucinate, use real logic.  
  
Features:  
- Shape recognition (tags, part types)  
- Auto-tagging + metadata fill  
- AI drawing assistant (suggest next steps)  
- BOM generation from shapes  
- Export BOM to PDF, Excel, JSON

# 💵 PHASE 4: SaaS-Ready Platform

Goals: Monetize + lock it down  
  
Backend:  
- User login/auth (Supabase or Firebase)  
- Project save/load  
- Stripe licensing tiers (Basic, Pro, Enterprise)  
- Deployment via Vercel/Netlify  
  
Packaging:  
- Export: PDF, SVG, PNG  
- Print-ready layouts  
- Embed AI Assistant module in canvas

# ☠️ Build Rules Stay in Effect

- No summary overwrites  
- No generative shortcuts  
- No hallucinated shapes or styles  
- Every phase tested before stacking next

# 🔗 CtrlSketch Pro + Parts Plus Integration Plan

Two-engine system: CtrlSketch (Frontend) + Parts Plus (Admin/DB Tool)

CtrlSketch Pro (Frontend):  
- Drawing UI  
- Part tagging  
- BOM generation  
- AI assistant  
- DB connection via API  
  
Parts Plus (Backend):  
- No canvas  
- Full part catalog with metadata  
- BOM validation + enrichment  
- Admin management tools  
- Own DB (Supabase/PostgreSQL)

Deployment Strategy:  
- Phase 1: Build CtrlSketch canvas with tagged shapes  
- Phase 2: Build Parts Plus microapp to manage part catalog  
- Phase 3: Connect CtrlSketch to Parts Plus via API for BOM enrichment

Folder Structure:

/ctrl-sketch-pro/ ← Vue frontend  
/parts-plus-api/ ← Node/Express backend or Supabase project

Bonus:  
- Run different teams or permission levels  
- Deploy Parts Plus privately or headlessly  
- Protect licensing and part IP outside of the main drawing app