

Lovable.dev 101: คู่มือเปลี่ยนไอเดียเป็น เว็บแอปพลิเคชันด้วย AI

(ฉบับผู้เริ่มต้น)

The Modern Builder's Playbook —
สร้างซอฟต์แวร์จริงโดยไม่ต้องเขียนโค้ดเอง



PROJECT:
LOVABLE.DEV PLAYBOOK

SCALE:
ISOMETRIC NTS

ENGINEER:
AI-ASSISTED

DRAWING NO:
001.00

STATUS:
DEPLOYED

เปลี่ยนบทบาทจาก 'ผู้ใช้' เป็น 'ผู้จัดการแอปพลิเคชัน'

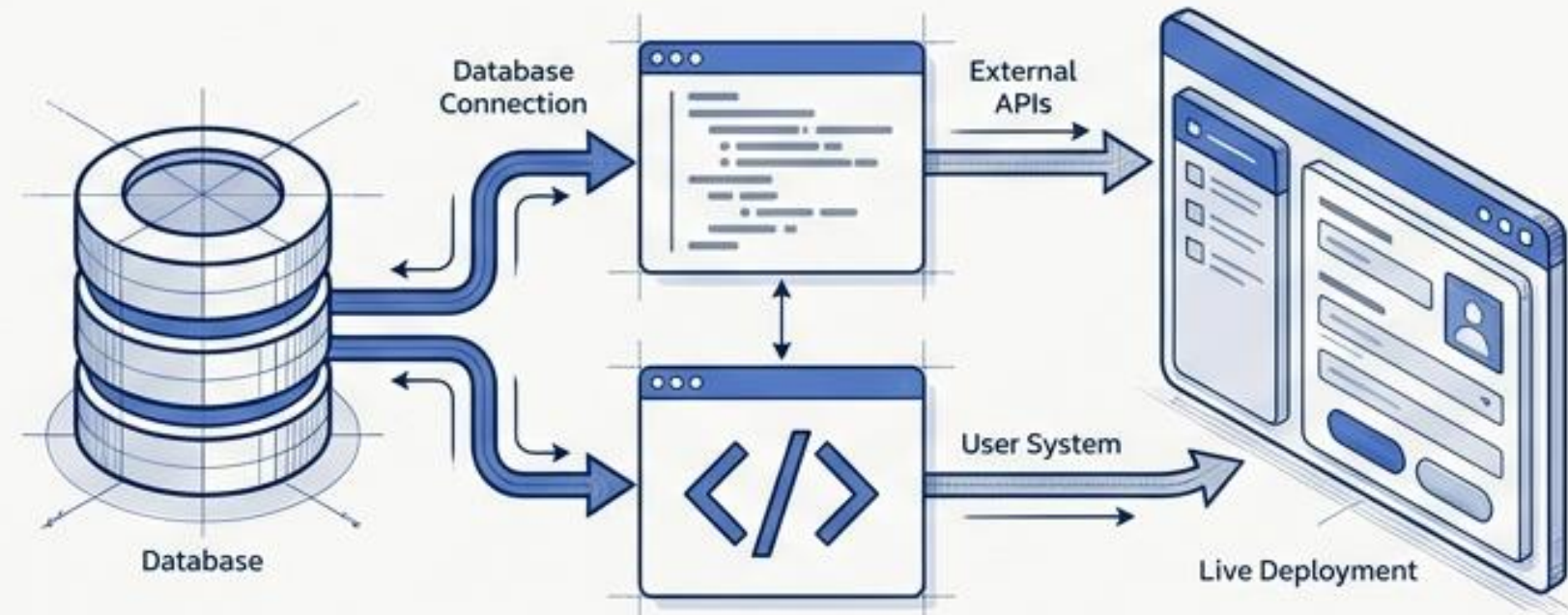
สิ่งที่คนมักเข้าใจผิด

สร้างแค่ภาพตัวอย่าง (Prototype)
หรือดีไซน์ชั่วคราว



พลังที่แท้จริงของ Lovable

เขียนโค้ดจริง (Real Code) เชื่อมต่อฐานข้อมูล ระบบสมาชิก
และ API ภายนอกได้ พร้อมนำขึ้นใช้งานจริง



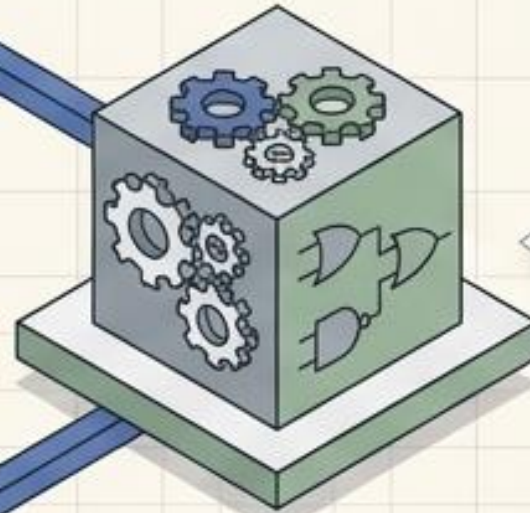
Footer Insight: AI คือ 'นักพัฒนา' ประจำตัวคุณ หน้าทีของคุณคือการกำหนดความต้องการ
ตรวจสอบผลลัพธ์ และพิจารณาความปลอดภัย

โครงสร้างทางสถาปัตยกรรมของ Web App



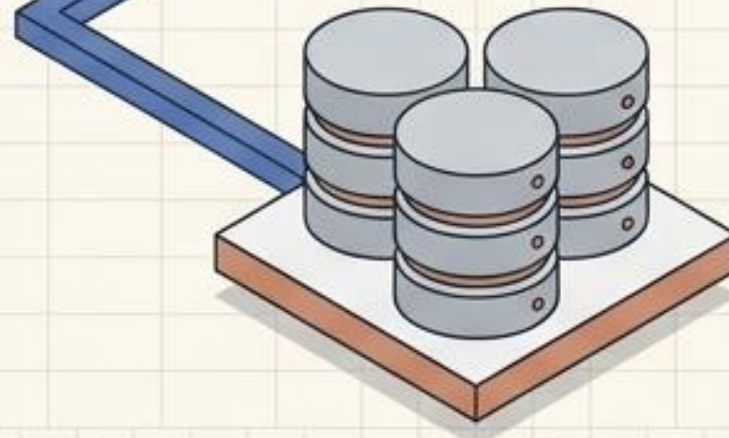
Frontend (ส่วนหน้าบ้าน)

ส่วนที่ผู้ใช้งานมองเห็นและโต้ตอบ เช่น หน้าแรก เมนู แบบฟอร์ม Dashboard และรองรับทุกหน้าจอ



Backend (เครื่องยนต์กลไก)

ระบบเบื้องหลัง เช่น ตรวจสอบรหัสผ่าน คำนวณ คะแนน ส่งอีเมล เช็คสิทธิ์ และเชื่อมต่อ API



Database (ฐานข้อมูล)

พื้นที่เก็บข้อมูลอย่างเป็นระบบ เช่น รายชื่อสมาชิก ประวัติการใช้งาน คำตอบแบบประเมิน

Blueprint 7 ประการ: สิ่งที่ต้องเตรียมก่อนพิมพ์ Prompt แรก



อย่าเริ่มด้วยคำสั่งสั้นๆ



ปัญหา (Problem)

- เช่น การตรวจฟอร์มแบบ
แบบนวนใช้เวลาานาน



ผู้ใช้ (Users)

- เช่น แอดมิน, สต๊าฟฟ์,
ผู้ใช้งานทั่วไป



งาน (Tasks)

- เช่น สมัคร, อนุมัติ,
ดาวน์โหลดเอกสาร



หน้าจอ (Screens)

- เช่น หน้าแรก, Dashboard,
หน้า Login



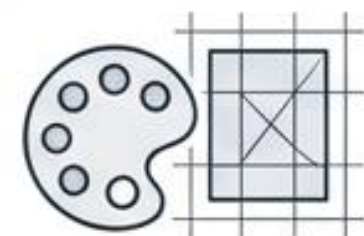
ข้อมูล (Data)

- เช่น ชื่อ, อีเมล,
สถานะ, วันที่



กฎ (Rules)

- เช่น รับจำกัด 60 คน,
แอดมินแก้ไขได้เท่านั้น



ดีไซน์ (Design)

- เช่น ธีมสี, สไตล์,
รองรับมือถือเป็นหลัก

สมการ Prompt สำหรับสถาปัตยกรรมแอป

[[Context บริบท] + [Goal เป้าหมาย] + [Users ผู้ใช้] + [Features ฟังก์ชัน] +
+ [Rules กฎ] + [Design ดีไซน์] + [Constraints ข้อจำกัด]]

ตัวอย่างการประกอบร่าง

สร้างระบบโรงเรียนอนุรักษสิ่งแวดล้อม (Context บริบท) เพื่อบันทึกกิจกรรม (Goal เป้าหมาย)... ผู้ใช้คือครูและนักเรียน (Users ผู้ใช้)... สร้างหน้า Dashboard (Features ฟังก์ชัน)... นักเรียนแก้ไขได้เฉพาะร.ร.ตัวเอง (Rules กฎ)... ธีมสีเขียว (Design ดีไซน์)... เริ่มจาก Frontend ยังไม่ต่อฐานข้อมูล (Constraints ข้อจำกัด)



Pro Tip: กำหนด Acceptance Criteria เสมอ (เกณฑ์ที่ใช้ตัดสินว่างานเสร็จ เช่น ค้นหาซื้อได้, แสดงผลบนมือถือได้)

วงจรการพัฒนา: 1 Prompt = 1 เป้าหมาย



หลีกเลี่ยงคำสั่ง 'ช่วยปรับให้สวยขึ้นและเพิ่มระบบทั้งหมด' ให้แยกคำสั่งย่อยเพื่อไม่ให้โครงสร้างพัง

สมรภูมิ Backend: เลือกเครื่องยนต์ให้แอปของคุณ



Lovable Cloud

- ✓ รวดเร็ว เรียบง่าย ไม่ต้องตั้งค่าบริการเอง
- ✓ **เหมาะสำหรับ:** ผู้เริ่มต้น, Prototype, แอปขนาดเล็กที่ต้องการความไว

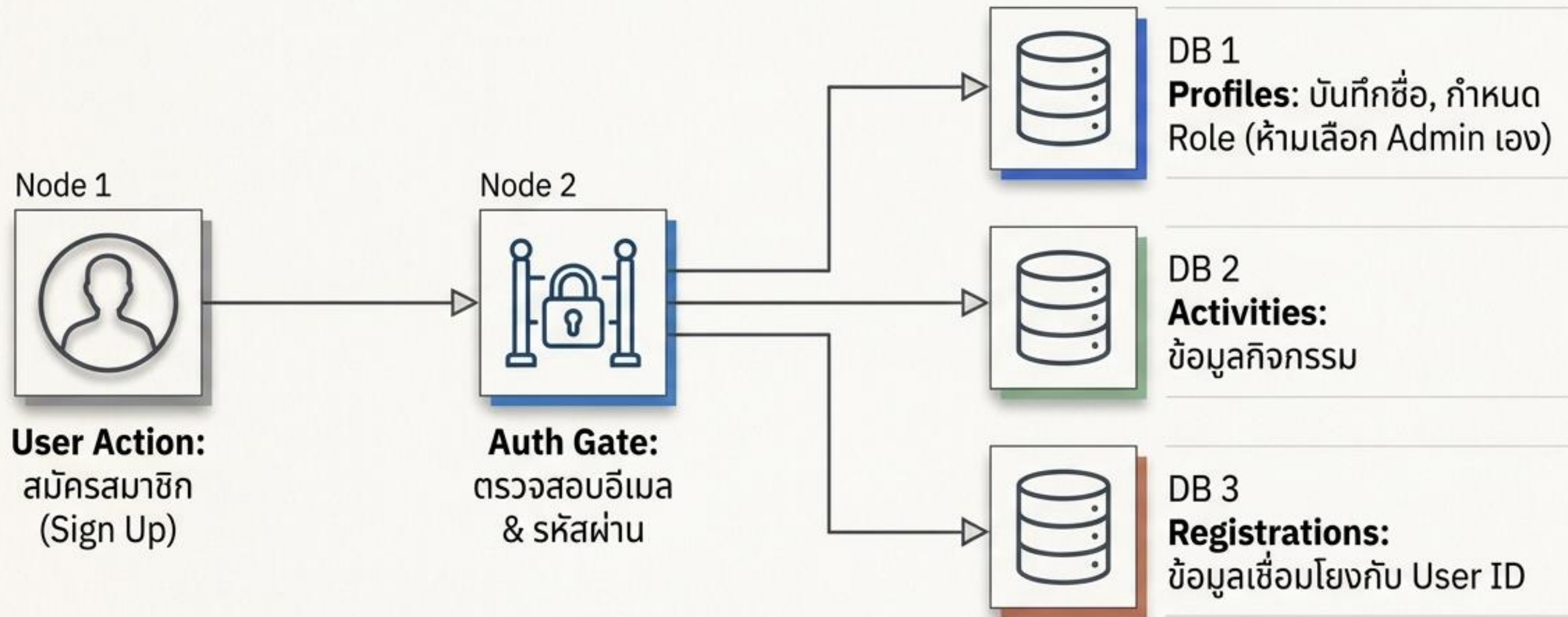


Supabase

- ✓ ฐานข้อมูล PostgreSQL, ระบบ Auth, การเก็บไฟล์, Row Level Security (RLS)
- ✓ **เหมาะสำหรับ:** แอปที่สเกลได้จริง, ต้องจัดการสิทธิ์ซับซ้อน, ใช้งานผ่าน Dashboard

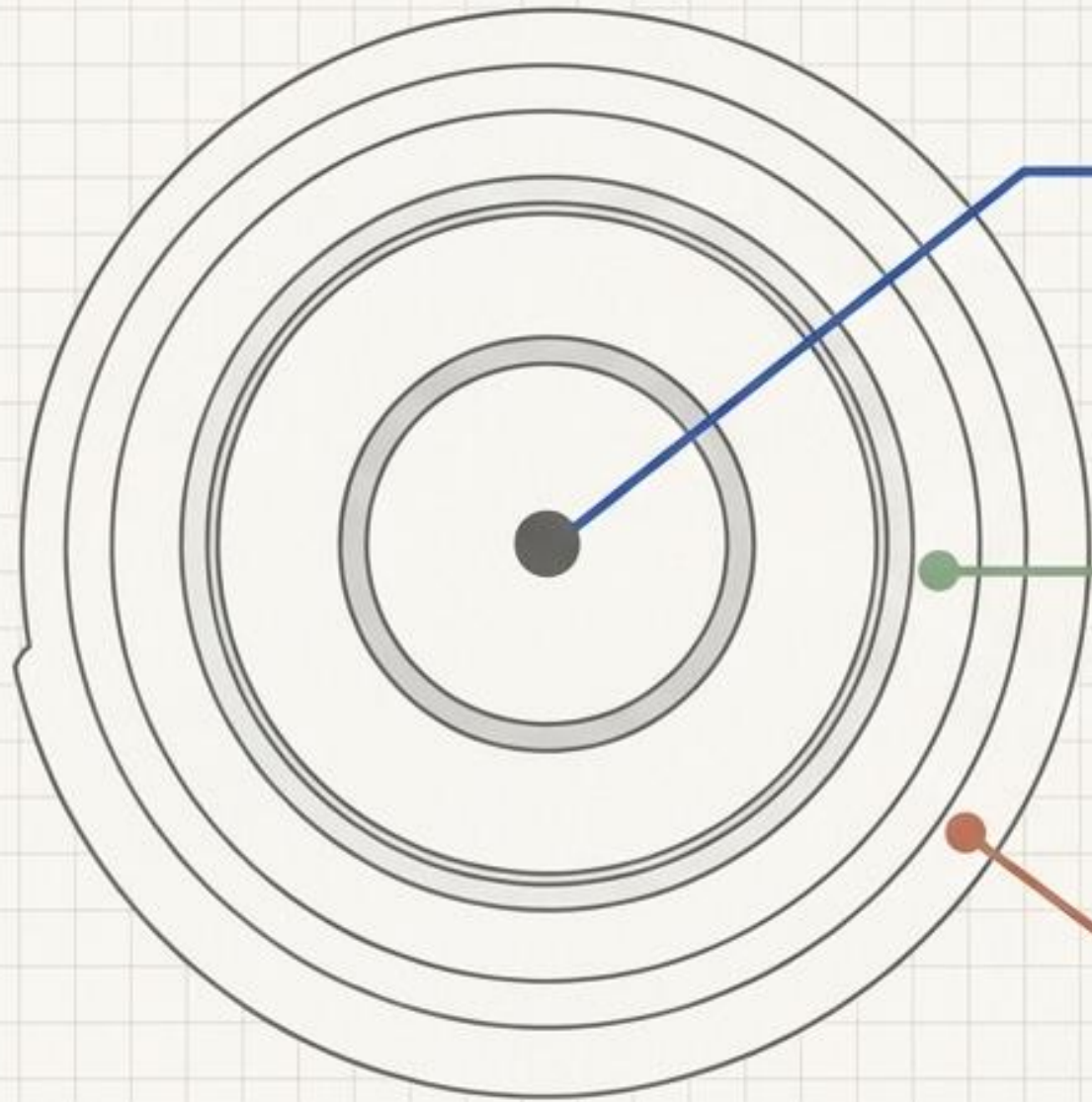
กฎเหล็ก: ทำ UI ให้เสร็จก่อนเชื่อม Backend เสมอ

สถาปัตยกรรมข้อมูล: การผสานระบบ Auth และ Supabase



Tip: สั่ง AI ให้ชัดเจน: 'สร้างตาราง Profiles, Activities, Registrations และกำหนด Schema ให้ถูกต้อง'

ไล่พิทักษ์ข้อมูล: แอปที่ทำงานได้ ไม่ได้แปลว่าปลอดภัย



1. No Secrets: ห้ามฝัง Secret Key หรือรหัสผ่านในโค้ดฝั่งผู้ใช้

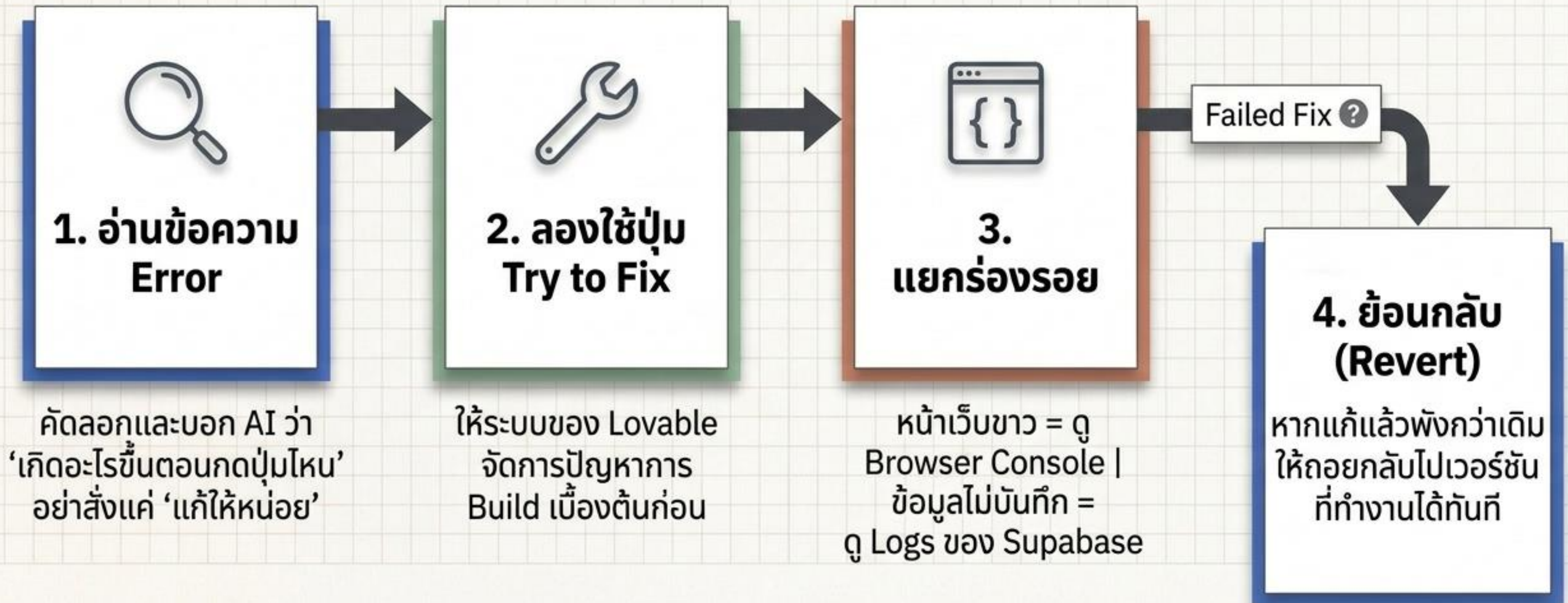
2. Logic (RLS): ล็อกที่ฐานข้อมูล ผู้ใช้เห็นและแก้ไขได้เฉพาะข้อมูลตัวเองเท่านั้น

3. Privacy (PDPA): เก็บข้อมูลที่จำเป็น มีระบบขอความยินยอมและลบข้อมูล



ข้อมูลสุขภาพ หรือบัตรประชาชน ต้องให้ผู้เชี่ยวชาญตรวจสอบก่อนใช้งานจริง

Bug Hunting Protocol: ระบบวิเคราะห์ข้อผิดพลาด



The Launchpad: จากโค้ดสู่โลกออนไลน์



Mindset ของผู้สร้าง: มือใหม่ vs. มือโปร

สิ่งที่มือใหม่พลาด

- ✗ สร้างระบบใหญ่ครบวงจรตั้งแต่เริ่ม
- ✗ เปลี่ยนความต้องการไปมาตลอดเวลา
- ✗ สั่งแก้ 10 เรื่องใน Prompt เดียว
- ✗ เชื่อมฐานข้อมูลทันทีที่เริ่มโปรเจกต์
- ✗ เชื่อว่า AI เขียนโค้ดถูก 100% เสมอ

นิสัยของมือโปร

- ✓ เริ่มจาก Core Feature (MVP) ก่อน
- ✓ กำหนดขอบเขต V1 ให้ชัดเจนแล้วค่อยขยาย
- ✓ สั่ง 1 Prompt ต่อ 1 เป้าหมาย (ใช้ Plan Mode)
- ✓ ทำ UI ให้จบด้วยข้อมูลจำลองก่อนต่อ Database
- ✓ ตรวจสอบสิทธิ์ (RLS) และ Security ทุกครั้ง

Roadmap 14 วัน: จากศูนย์สู่ผู้สร้าง

Days 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Days 1-4: โครงสร้างและ UI

เรียนรู้ Layout, Responsive Design
และสร้างเว็บ 4-5 หน้า (ยังไม่มีฐานข้อมูล)

Days 5-7: ฟอर्मและตรรกะ

สร้างระบบคำนวณ, แบบฟอร์ม, และจงใจ
ทดสอบกรอกข้อมูลผิดเพื่อฝึกแก้ Bug

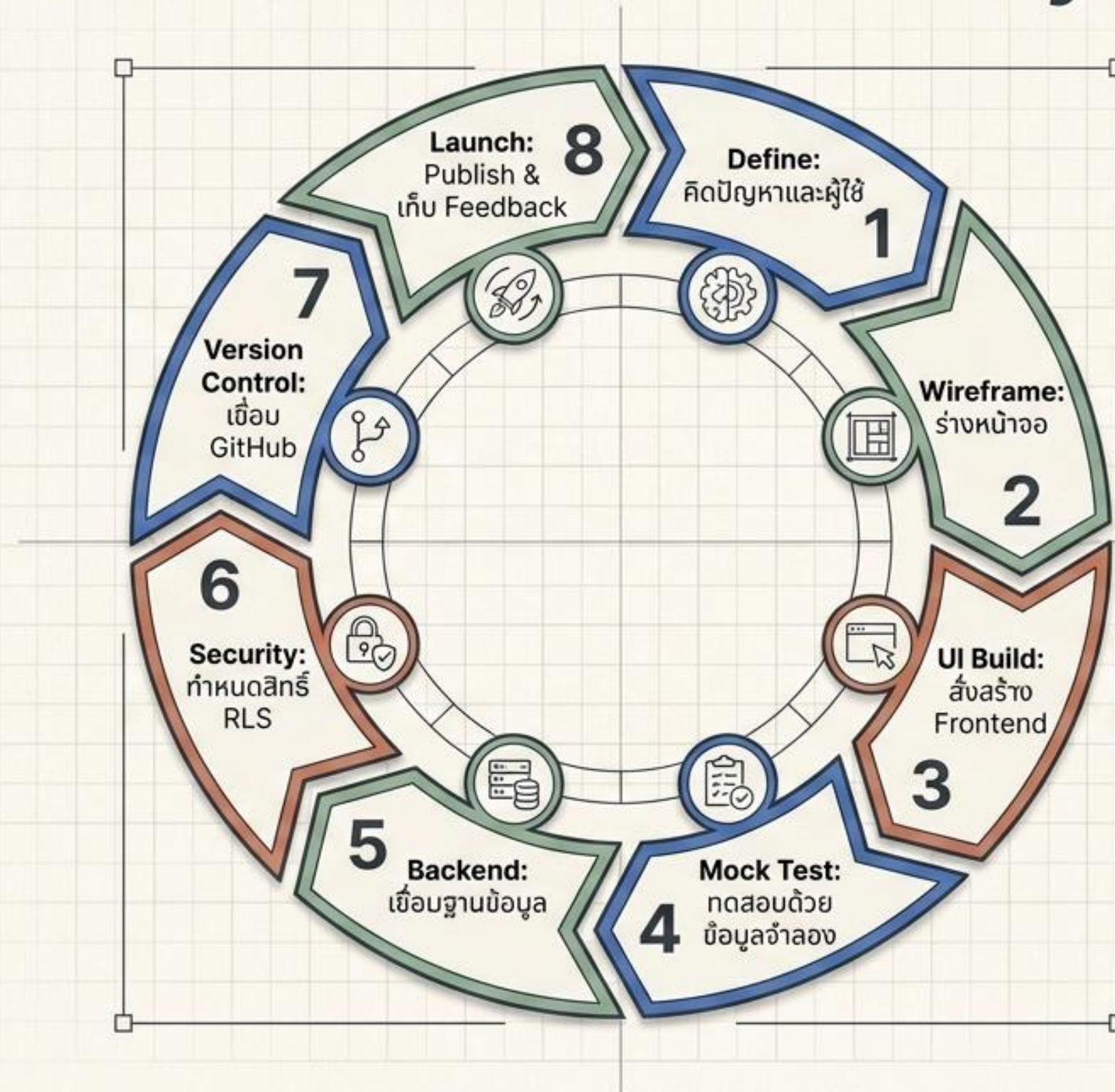
Days 8-11: ฐานข้อมูลและ Auth

เชื่อม Supabase, สร้างตาราง (CRUD),
เพิ่มระบบ Login, และตั้งค่า RLS

Days 12-14: ควบคุมและเผยแพร่

เชื่อม GitHub (Commit/Sync), ใช้ Test
Checklist ตรวจสอบ, และ Publish!

The Universal Creator Cycle



ภารกิจแรกของคุณ (Your First Mission)



ภารกิจแรกของคุณ
แบบประเมินความสัมพันธ์
30 ข้อ

โฟกัส: ตรรกะ (Logic), แบบฟอร์ม,
การคำนวณคะแนน, และหน้าผลลัพธ์

ความยาก: ระดับเริ่มต้น (ฝึกความแม่นยำของ
UI)

Start



ภารกิจที่สอง
ระบบลงทะเบียนกิจกรรม

โฟกัส: Full-Stack, ระบบ Login, ฐานข้อมูล
(Supabase), และ Admin Dashboard

ความยาก: ระดับกลาง (ฝึกเชื่อมต่อข้อมูลและ
จัดการสิทธิ์)

Start

เปิด Lovable.dev แล้วเริ่มพิมพ์ Prompt แรกของคุณได้เลย!