AI AGENT CO-WORKING SYSTEM · DAY 1

Basic CRM System with Vibe Coding

ออกแบบระบบ CRM ให้ตอบโจทย์การใช้งานของคุณด้วย AI Agent

เช้าเข้าใจการพัฒนาระบบและ AI Agent แล้วออกแบบระบบเป็น CLAUDE.md ไฟล์เดียว · บ่ายสร้างทีละเฟส จนได้ CRM ที่สมัคร ล็อกอิน และแบ่งสิทธิ์ 3 บทบาทได้จริง



STEP 01

เข้าใจ

SDLC · ระบบ 3 ชั้น · AI Agent ทำงานจริงอย่างไร

STEP 02

ออกแบบ

Use Case · Permission Matrix · CLAUDE.md แบ่ง 3 เฟส

STEP 03

Phase 1

สมัคร + ล็อกอิน 3 บทบาท + อนุมัติทีมงานใหม่

STEP 04

Phase 2

ลิด CRM ล็อกสิทธิ์ถึงฐานข้อมูล

GETTING STARTED

วิธีใช้คู่มือเล่มนี้

คู่มือเรียงตามลำดับของวันเรียน ช่วงเช้าเข้าใจว่าระบบถูกพัฒนาอย่างไรและ AI Agent ทำงานจริงอย่างไร แล้วออกแบบระบบลงในไฟล์ `CLAUDE.md` ไฟล์เดียว ช่วงบ่ายสั่ง Agent สร้างทีละเฟสจากไฟล์นั้น ทุกขั้นมีคำอธิบายสั้น ๆ และ Prompt สำเร็จรูปที่คัดลอกไปวางใน Claude Code หรือ Codex ได้ทันที

ปุ่มคัดลอก
ในคู่มือฉบับเว็บ ทุกกล่อง Prompt มีปุ่ม "คัดลอก Prompt" มุมขวาบน

[ ] ตัวแดงในวงเล็บ
สิ่งที่ต้องแนบหรือเปลี่ยนเองก่อนกดส่ง

ด่านทดสอบ
สิ่งที่ต้องเห็นบนจอจริงก่อนไปเฟสถัดไป

กล่องแดงอ่อน
หลักคิดหรือข้อควรระวังที่สำคัญ

LEARNING OUTCOMES

ผลลัพธ์หลังเรียน

จบวันนี้ ทีมสโมสร (เจ้าของ/ผู้จัดการ · แอดมินและดูแลสมาชิก · ฝ่ายขายและการตลาด) จะเข้าใจว่าระบบ 1 ระบบถูกสร้างขึ้นอย่างไร และได้ระบบ CRM ต้นแบบที่ใช้งานได้จริงกลับไป — ไม่ต้องมีพื้นฐานเขียนโปรแกรม

- ✓ **อธิบายได้** ว่าการพัฒนาระบบมี 6 ขั้น (SDLC) และ 1 ระบบประกอบด้วย Frontend · Backend · Database
- ✓ **แยก Authentication กับ Authorization ได้** และรู้ว่าทำไมต้องล็อกสิทธิ์ถึงฐานข้อมูล (Row Level Security)
- ✓ **เข้าใจการทำงานจริงของ Main Agent / Subagent** และเหตุผลที่คลาสนี้ใช้ Main Agent ตัวเดียว
- ✓ **ออกแบบระบบเป็น CLAUDE.md ไฟล์เดียว:** Use Case · ตารางสิทธิ์ · ตารางข้อมูล · แผนงาน 3 เฟส
- ✓ **ได้ระบบ CRM ต้นแบบ (Next.js + Supabase) รันบนเครื่อง:** สมัคร + อนุมัติทีมงานใหม่ + ล็อกอิน 3 บทบาท (Phase 1) และลีด CRM (Phase 2)
- ✓ **ทำต่อที่บ้านได้เอง:** Dashboard ผู้บริหาร + ขึ้นเว็บจริงบน Cloudflare (Phase 3)

ชิ้นงานที่ได้ระหว่างวัน

`Use Case` `Permission Matrix` `CLAUDE.md (Phase 1-3)` `CRM Phase 1 + 2` `Git จุดย้อนกลับ`

> **ขอบเขตของคลาส**
> เพื่อให้ผู้เรียนเข้าใจพื้นฐานอย่างแท้จริงภายใน 1 วัน คลาสนี้สอนเฉพาะแกนกลางของระบบ ส่วนที่เหลือผู้เรียนต่อยอดเองได้ด้วยวิธีเดียวกัน

DAY OVERVIEW

ตารางสอน

1 วัน 10:00–17:00 น. · เช้า = เข้าใจ + ออกแบบ · บ่าย = สร้างทีละเฟสจาก CLAUDE.md

เวลา	ช่วง	หัวข้อ	ผลลัพธ์ / ด่าน
10:00–10:45	**Session 1**	การพัฒนาระบบทำงานอย่างไร: SDLC · ระบบ 3 ชั้น (ร้านอาหาร) · ตามรอย 1 คำขอ · Authentication / Authorization	อธิบายระบบสโมสรเป็น 3 ชั้นได้
10:45–11:30	**Session 2**	AI Agent ทำงานจริงอย่างไร: ความเชื่อผิด vs ความจริง · Main Agent / Subagent · ทำไมใช้ Main Agent ตัวเดียว · Agent 2 แบบ · CLAUDE.md คืออะไร	รู้ว่า CLAUDE.md ที่ดี = งานที่ถูกต้อง
11:30–12:30	**Workshop 1**	เตรียม Env + ออกแบบระบบ: Claude Code · Git · Node.js · Cloudflare + Supabase · Use Case · หน้าจอตามบทบาท · Permission Matrix · เขียน CLAUDE.md	เครื่องพร้อม · CLAUDE.md แบ่ง 3 เฟส
12:30–13:30	พักกลางวัน		
13:30–15:00	**Workshop 2**	Phase 1 · สมัคร + ล็อกอิน + 3 บทบาท: วางคีย์ · PROMPT P1-BUILD	**ด่าน 1** รออนุมัติจริง · 3 บทบาทเห็นเมนูต่างกัน
15:00–15:15	เบรก		
15:15–16:30	**Workshop 3**	Phase 2 · ลีด CRM: PROMPT P2-BUILD ทีละชั้น Database → Backend → Frontend	**ด่าน 2** ลีดเข้า · เห็น/แก้ได้ตรงตารางสิทธิ์
16:30–17:00	**ปิดวัน**	สรุป + Phase 3 การบ้าน (PROMPT P3-BUILD: Dashboard + ขึ้นเว็บ) + ถาม-ตอบ	รู้ว่าทำต่อที่บ้านอย่างไร

> **หมายเหตุถึงผู้เรียน**
> ตารางเวลาเป็นแนวทาง ผู้สอนอาจปรับเวลาแต่ละช่วงให้เหมาะกับความคืบหน้าของผู้เรียน โดยเนื้อหาและผลลัพธ์หลักยังครบถ้วน

BEFORE CLASS

ก่อนวันเรียน

เตรียมบัญชีและเครื่องให้พร้อม ส่วนการติดตั้งเครื่องมือที่เหลือ ให้ Agent ช่วยทำใน Workshop 1

- ✓ **คอมพิวเตอร์ + เบราว์เซอร์ Chrome**
- ✓ **บัญชี Claude แบบรายเดือน ขั้นต่ำ Pro**
  หรือ ChatGPT Plus ถ้าใช้ Codex
- ✓ **ติดตั้ง Claude Code หรือ Codex**
- ✓ **อีเมลสำหรับสมัคร Supabase และ Cloudflare**
  สมัครฟรี ทำพร้อมกันใน Workshop 1

กติกา 3 ข้อของห้องเรียน

ใช้ข้อมูลจำลอง
ข้อมูลทั้งหมดในคลาสเป็นข้อมูลจำลอง · สโมสรตัวอย่างชื่อสมมติ **Harmony Club**

คีย์ลับห้ามอยู่ใน public/
ไฟล์ใน `public/` และโค้ดหน้าเว็บ ใครก็เปิดดูได้ คีย์ลับอยู่ใน `.env.local` ใช้ฝั่ง server เท่านั้น

คนอนุมัติเรื่องสำคัญ
AI ร่างและเขียนโค้ดได้ คนเป็นคนตรวจและตัดสินใจ

CLAUDE CODE หรือ CODEX

ใช้แอปไหนก็ได้ Prompt เดียวกัน

Prompt ทุกตัวในคู่มือใช้ได้ทั้งสองแอป ความต่างมีเท่านี้

เรื่อง	Claude Code	Codex
ไฟล์คู่มือโปรเจกต์	`CLAUDE.md`	`AGENTS.md` (เนื้อหาเดียวกัน)
บัญชีขั้นต่ำ	Claude Pro $20/เดือน	ChatGPT Plus
การถามคำถาม	ตัวเลือกให้กดเลือก	ถาม-ตอบเป็นข้อ ๆ ในแชต
แนบไฟล์/รูป	ลากไฟล์เข้าแชต หรือพิมพ์ path	ลากไฟล์เข้าแชต หรือพิมพ์ path

---

SESSION 1 · 10:00–10:45

S1 การพัฒนาระบบทำงานอย่างไร

ทุกระบบผ่าน 6 ขั้นเดียวกัน — **คนคิด ออกแบบ และตรวจ** ส่วน AI Agent ช่วยเขียนโค้ดในขั้นพัฒนา



01	02	03	04	05	06
เก็บรวบรวมความต้องการ + วางแผน	คิดวิเคราะห์ + ออกแบบ	พัฒนา	ทดสอบ	ขึ้นระบบ	ดูแล

วันนี้อยู่ขั้นไหน

ช่วงเช้า

คิดวิเคราะห์ + ออกแบบ

เข้าใจโจทย์ วาง Use Case ตารางสิทธิ์ แล้วเขียนเป็น CLAUDE.md

ช่วงบ่าย

พัฒนา + ทดสอบ

สั่ง Agent สร้างทีละเฟส แล้วทดสอบผ่านด่านทีละด่าน

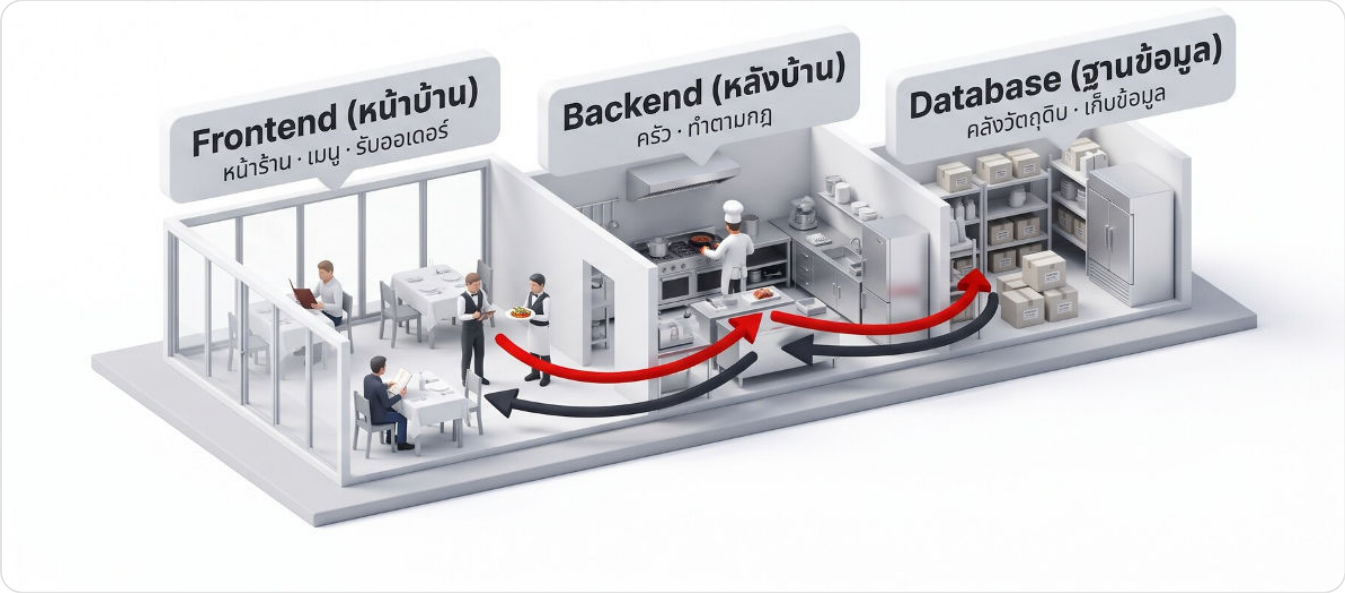
การบ้าน

ขึ้นระบบ

Phase 3 ขึ้นเว็บจริงบน Cloudflare แล้วทดสอบบนมือถือ

ระบบ 3 ชั้น · ร้านอาหาร

การทำงานของระบบ



Frontend (หน้าร้าน)

สิ่งที่ผู้ใช้เห็นและกด — หน้าสมัคร หน้าล็อกอิน หน้ารายการลีด

Backend (ครัว)

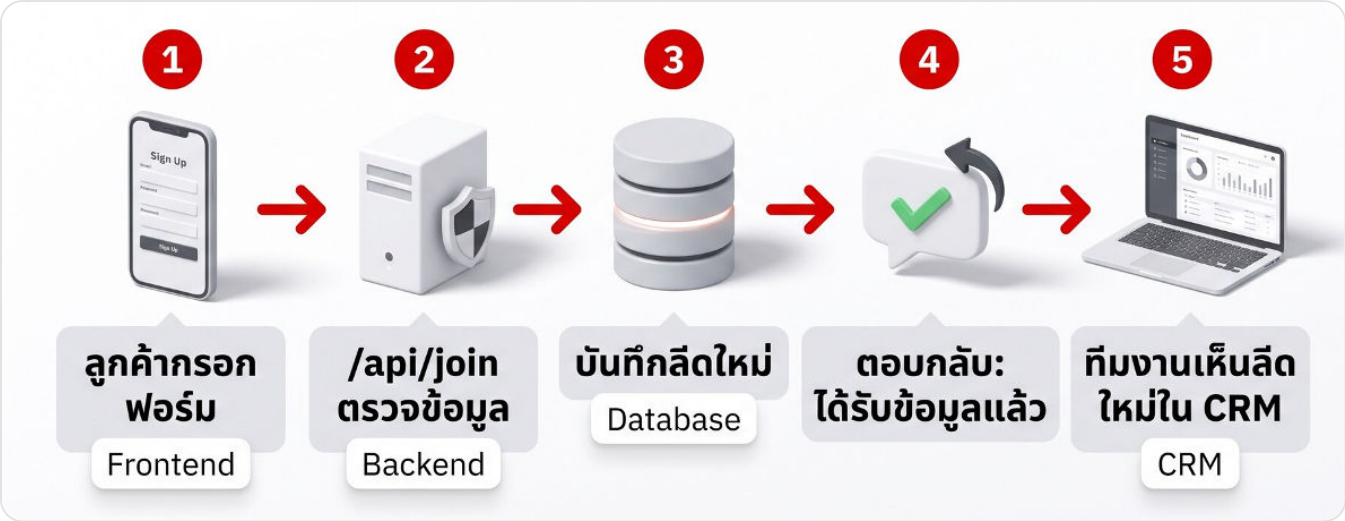
รับคำขอ ตรวจ ทำตามกฎ และเก็บคีย์ลับ — เช่น `/api/join` (Route Handler ของ Next.js) รับฟอร์มสนใจ

Database (คลังวัตถุดิบ)

เก็บข้อมูลถาวรเป็นตาราง — ผู้ใช้ ลีด ประวัติการติดต่อ

ตามรอย 1 คำขอ

ตัวอย่างการส่งข้อมูล



1. ลูกค้ากรอกฟอร์มสนใจบนหน้าเว็บ **(Frontend)**
2. ส่งไป `/api/join` ตรวจข้อมูล **(Backend)**
3. บันทึกเป็นลีดใหม่ในตาราง `leads` **(Database)**
4. ตอบกลับลูกค้าว่า "ได้รับข้อมูลแล้ว"
5. ทีมงานเปิด CRM เห็นลีดใหม่

AUTHENTICATION · AUTHORIZATION



Authentication

คุณคือใคร

ตรวจอีเมล + รหัสผ่าน (Supabase Auth ทำให้) ผ่านแล้วได้ "บัตรผ่าน" (session) ติดตัวไปทุกคำขอ

Authorization

คุณทำอะไรได้

อ่านบทบาทจากตาราง `profiles` → เปิดเมนูตามบทบาท และล็อกที่ฐานข้อมูลด้วย Row Level Security

ตัวอย่างในระบบสโมสร

สมัครแล้วแต่ยังรออนุมัติ = Authentication ผ่าน แต่ Authorization ยังไม่ให้เข้า · พนักงานขายล็อกอินได้ แต่ลบลีดไม่ได้

SESSION 2 · 10:45–11:30

S2 AI Agent ทำงานจริงอย่างไร



ความเชื่อผิด

Agent หลายตัวคุยกันเอง นัดแบ่งงานกันเอง

ความจริง

Agent ทำงานผ่าน ใบสั่งงาน · CLAUDE.md · ไฟล์ในโปรเจกต์



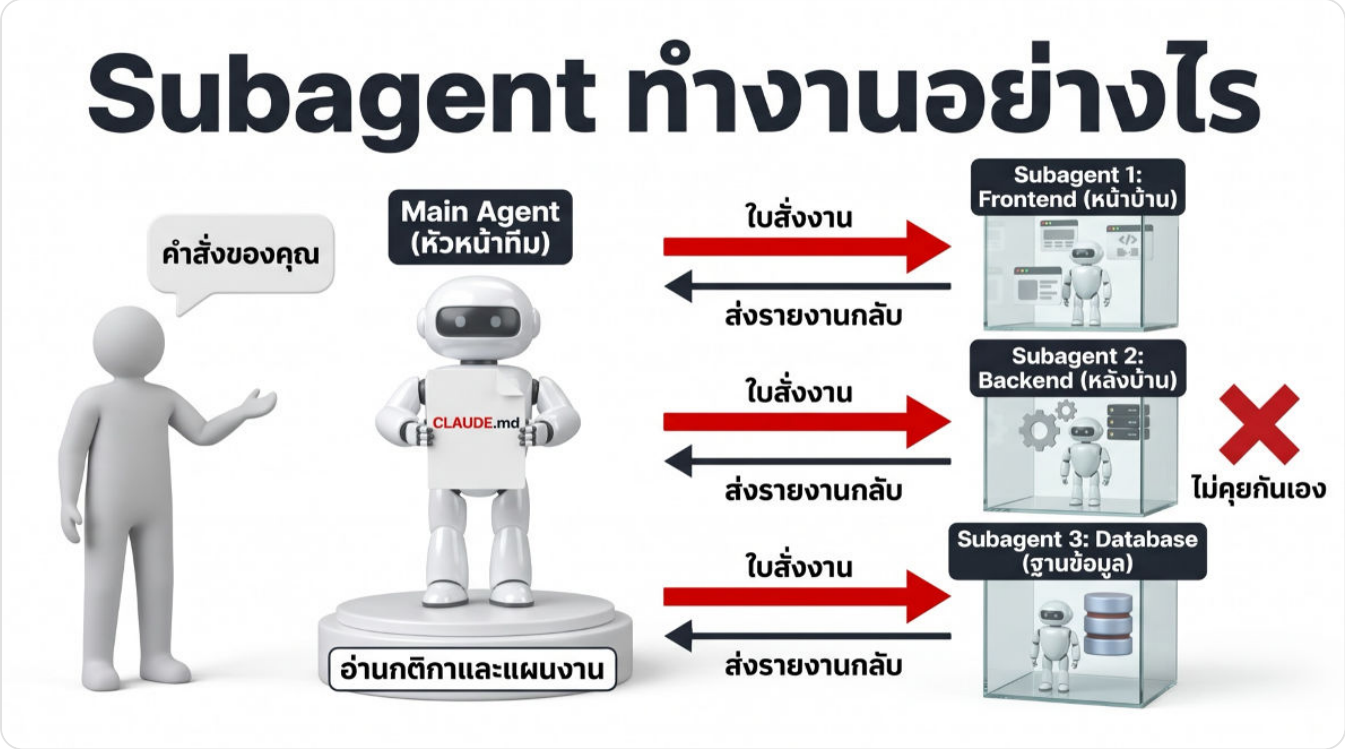
1. **Main Agent = หัวหน้าทีม**
   คุยกับเรา อ่านโจทย์ แล้วตัดสินใจว่าจะแตกงานไหม

2. **หัวหน้าเขียนใบสั่งงาน (brief)**
   ให้ Subagent แต่ละตัว

3. **Subagent เริ่มจากความจำว่าง**
   ไม่เห็นบทสนทนาของเรา รู้แค่ใบสั่งงาน + ไฟล์กติกา (CLAUDE.md) + ไฟล์ในโปรเจกต์

4. **ทำเสร็จส่งรายงานกลับหัวหน้าเท่านั้น**
   Subagent ด้วยกันไม่คุยกัน

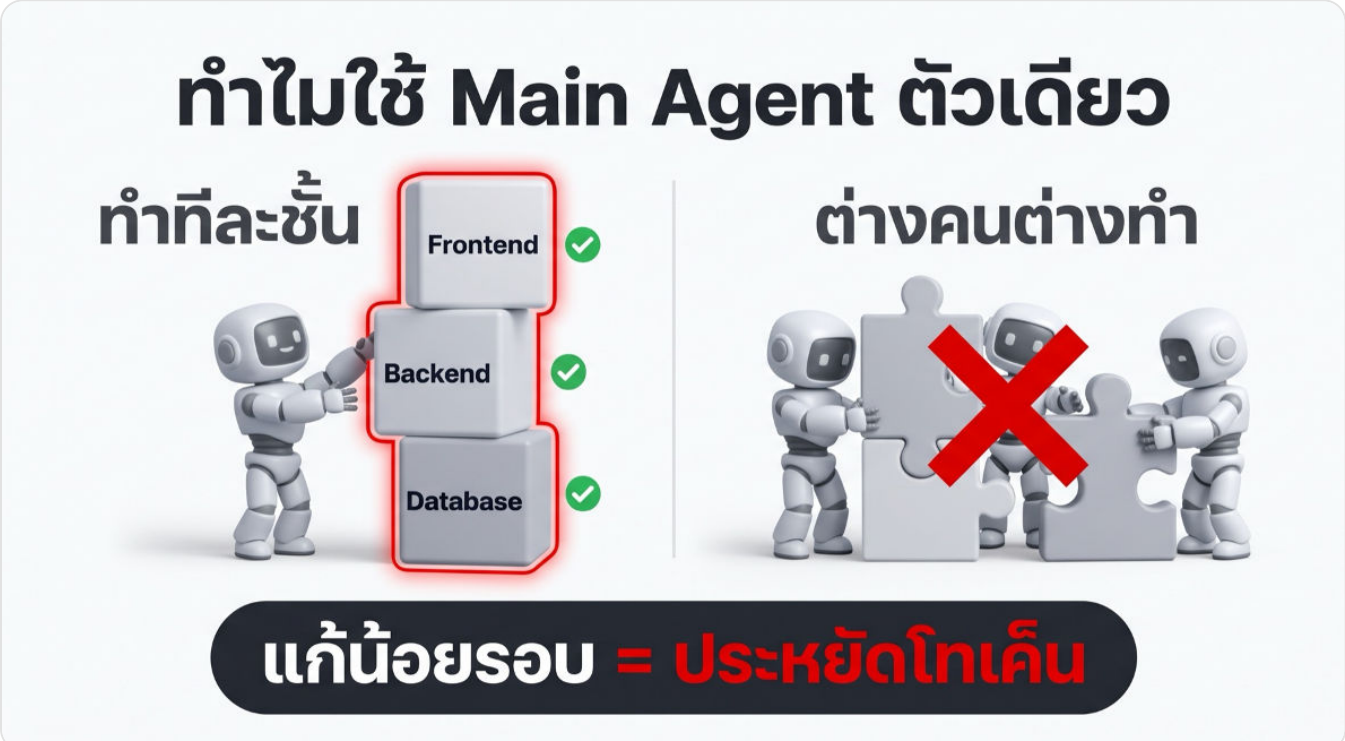
5. **หัวหน้ารวมผล ตรวจ แล้วรายงานเรา**



หมายเหตุความถูกต้อง

มีระบบแบบทีม (เช่น agent teams) ที่ส่งข้อความหากันได้ แต่เป็นช่องทางที่ออกแบบไว้ และเริ่มจากคำสั่งของคนหรือหัวหน้าเสมอ — ไม่ใช่ AI ตื่นมาประชุมกันเอง

WHY ONE MAIN AGENT



1 **ทุกชั้นพึ่งกัน**

Frontend เรียก Backend · Backend ใช้ตารางของ Database — แยกกันทำ ต่างคนต่างตัดสินใจ (ชื่อคอลัมน์ รูปแบบข้อมูล) รวมแล้วไม่เข้ากัน

2 **โทเค็นแพงขึ้นมาก**
ระบบหลาย Agent ใช้โทเค็นราว 15 เท่าของแชตปกติ (Agent ตัวเดียวราว 4 เท่า)

3 **ที่แพงจริงคือจำนวนรอบที่ต้องสั่งแก้**
ทำพร้อมกันเสร็จเร็วขึ้น แต่ไม่ได้ถูกต้องขึ้น

4 **เริ่มจากแบบที่ง่ายที่สุด**
แล้วค่อยเพิ่มความซับซ้อนเมื่อจำเป็นเท่านั้น

5 **Subagent ใช้ช่วยอ่าน ค้น ตรวจ**
ไม่ใช่เขียนหลายชั้นของระบบเดียวพร้อมกัน

Subagent เหมาะกับ	ไม่เหมาะกับ
ค้นหา / อ่านโค้ดแล้วสรุป · รีวิวหรือตรวจทาน · งานที่แยกขาดกันจริง	เขียนหลายชั้นของระบบเดียวพร้อมกัน · งานที่ต้องโต้ตอบบ่อย · งานเล็กแก้จุดเดียว

อ้างอิง Anthropic · Multi-agent research system · Anthropic · Building effective agents · Cognition · Don't Build Multi-Agents · Claude Code Docs · Subagents

TWO KINDS OF AGENT



	Agent ช่วยสร้างระบบ	Agent ที่ทำงานในระบบ
ตัวอย่าง	Claude Code / Codex	แชตบอต · ผู้ช่วยสรุปลีด
ทำงานเมื่อไหร่	เมื่อคนสั่ง	ตลอด 24 ชม.
อะไรสั่งให้เริ่ม	คนพิมพ์คำสั่ง	เหตุการณ์ เช่น ลีดใหม่เข้า
คนอยู่ตรงไหน	ตรวจและอนุมัติโค้ด	กดอนุมัติก่อนส่งถึงลูกค้า
ในคลาสนี้	**ใช้วันนี้**	งานต่อยอด

CLAUDE.md คืออะไร

คู่มือโปรเจกต์ที่ Agent อ่านเองทุกครั้งที่เปิดงาน — ไฟล์เดียวที่ root ของโปรเจกต์ รวมทุกอย่างที่ Agent ต้องรู้ แทนการอธิบายซ้ำในแชตทุกครั้ง

ในไฟล์มีอะไร
เป้าหมาย · บทบาท · สิทธิ์ · ข้อมูล · กติกา · แผนงาน 3 เฟส

สั่งงานสั้นลง
เหลือแค่ `อ่าน CLAUDE.md แล้วทำ Phase 1`

เขียนให้กระชับ
ไม่เกิน ~150 บรรทัด ถ้าโปรเจกต์ใหญ่ขึ้นค่อยแยกรายละเอียดไปไว้ใน `docs/` แล้วอ้างถึงจาก CLAUDE.md

ใช้ Codex
ตั้งชื่อไฟล์ `AGENTS.md` เนื้อหาเดียวกัน

CLAUDE.md มี 3 ระดับ

Global
`~/.claude/CLAUDE.md`
นิสัยการทำงานของเราทุกโปรเจกต์
Codex: `~/.codex/AGENTS.md`

Project · วันนี้เขียนระดับนี้
`./CLAUDE.md`
ของโปรเจกต์นี้ แชร์กับทีมผ่าน Git
Codex: `./AGENTS.md`

Subproject
`CLAUDE.md` ในโฟลเดอร์ย่อย
อ่านเมื่อทำงานในโฟลเดอร์นั้น
Codex: `AGENTS.md` ในโฟลเดอร์ย่อย

บทเรียน
CLAUDE.md ที่ดี = ใบสั่งงานที่ดี = งานที่ถูกต้อง — เช้านี้เราจะเขียนไฟล์นี้ด้วยกันใน Workshop 1

อ้างอิง Claude Code Docs · Memory (CLAUDE.md)

WORKSHOP 1 · 11:30–12:30

W1 เตรียม Env + ออกแบบระบบ

เป้าหมาย: เครื่องพร้อมทำงาน · โปรเจกต์ Next.js รันบนเครื่อง · มีโปรเจกต์ Supabase และบัญชี Cloudflare · ได้ไฟล์ `CLAUDE.md` ที่แบ่งงานเป็น Phase 1–3



1.1 เครื่องมือวันนี้

Claude Code
Main Agent ช่วยเขียนโค้ด · ใช้กับบัญชี Claude แบบรายเดือน ขั้นต่ำ Pro $20/เดือน (หรือ Codex กับ ChatGPT Plus)

Git
Version Control บันทึกจุดย้อนกลับทุกครั้งที่ผ่านด่าน

Node.js
ใช้สร้างและรันโปรเจกต์ Next.js (`npm run dev`) และขึ้นเว็บ

ไม่ต้องติดตั้งเอง — ให้ Agent ตรวจเครื่อง ติดตั้งตัวที่ขาด สร้างโปรเจกต์ Next.js `club-crm` ด้วย create-next-app (App Router · TypeScript · Tailwind) ติดตั้ง `@supabase/ssr` แล้วเปิด `npm run dev` ดูที่ `http://localhost:3000`

PROMPT SETUP — เตรียม Env · ตรวจเครื่อง + สร้างโปรเจกต์ Next.js

```
ช่วยเตรียมเครื่องของฉันสำหรับทำระบบ CRM วันนี้:
1. ตรวจว่ามี Git และ Node.js (เวอร์ชัน 20 ขึ้นไป) หรือยัง ตัวไหนยังไม่มีให้ช่วยติดตั้ง (ถามก่อนทุกครั้งที่ต้องใช้รหัสผ่านเครื่อง)
2. สร้างโปรเจกต์ Next.js ชื่อ club-crm ด้วยคำสั่ง
   npx create-next-app@latest club-crm --ts --app --tailwind --eslint --no-src-dir --import-
alias "@/*" --use-npm --yes
3. ติดตั้ง Supabase สำหรับ Next.js: npm install @supabase/supabase-js @supabase/ssr
4. ตรวจว่า .gitignore กันไฟล์ลับ .env* แล้ว และ Git เริ่มแล้ว
5. รัน npm run dev แล้วบอกให้ฉันเปิด http://localhost:3000 ดูว่าขึ้นหน้าเว็บ
สรุปผลเป็นตาราง: รายการ · สถานะ · ต้องทำอะไรต่อ
```

1.2 Cloudflare + Supabase

ระบบของเราใช้บริการฟรี 2 ตัว — Supabase ทำหน้าที่ฐานข้อมูลและระบบล็อกอิน ส่วน Cloudflare เป็นที่ฝากเว็บจริงใน Phase 3 — ระหว่างเรียนเรารันบนเครื่องด้วย `npm run dev`

Supabase

ฐานข้อมูล + ระบบล็อกอิน

Database · Authentication · Row Level Security

1. **New project** ชื่อ `club-crm`
2. ตั้งรหัสผ่านฐานข้อมูล แล้วบันทึกเก็บไว้
3. เลือก Region **Singapore**
4. **Authentication → Sign In / Providers → Email** ปิด "Confirm email" (สำหรับคลาส)
5. **Project Settings → API Keys** จด Project URL + publishable key (secret key ใช้ Phase 2)

Cloudflare

ที่ฝากเว็บจริง

Workers + OpenNext adapter สำหรับ Next.js

1. สมัครฟรีที่ `dash.cloudflare.com` ไว้ก่อน
2. ใช้จริงตอน Phase 3 (การบ้าน) — Agent ตั้งค่าให้

ใช้เมื่อไหร่: ระหว่างเรียนรันบนเครื่องด้วย `npm run dev` · ขึ้นเว็บจริงใน Phase 3 ด้วย `npm run deploy`

publishable key

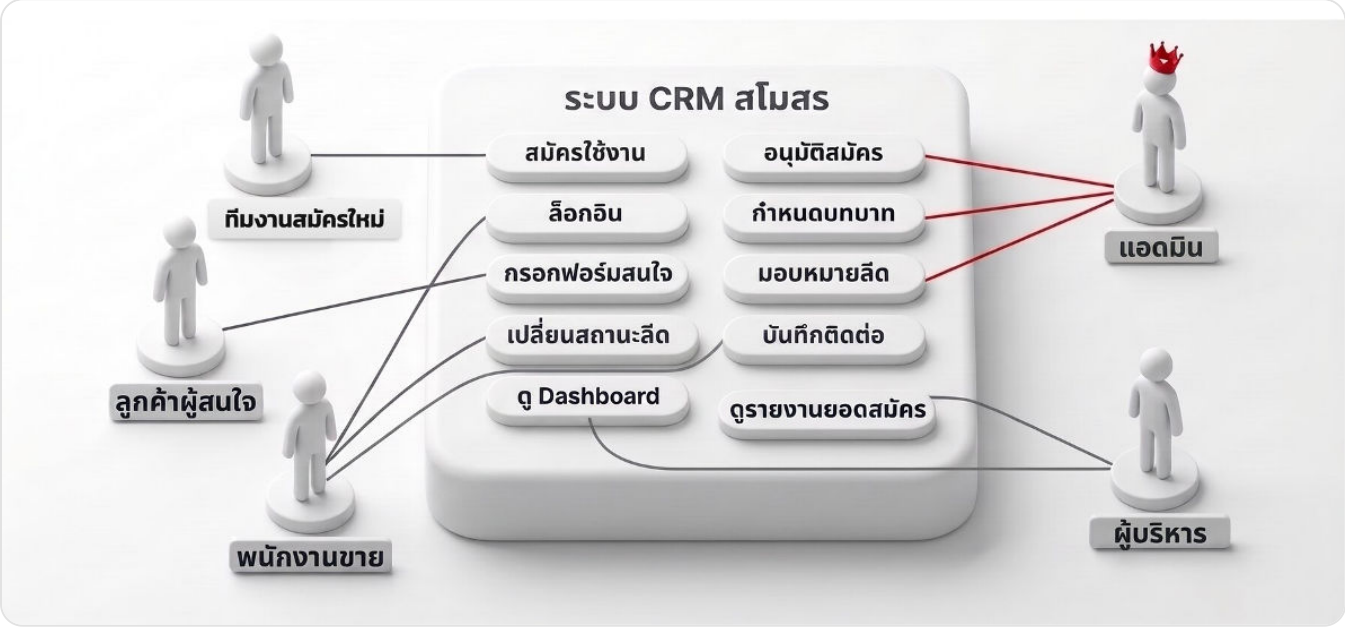
ใส่หน้าเว็บได้ (`.env.local` · `NEXT_PUBLIC_SUPABASE_PUBLISHABLE_KEY`)

secret key

= กุญแจห้องเก็บของ **ห้ามใส่หน้าเว็บหรือวางในแชต** · ใช้ฝั่ง server เท่านั้น (`SUPABASE_SECRET_KEY`)

1.3 Use Case — ใครใช้ระบบทำอะไร

ผู้กระทำ 5 แบบ กับการกระทำ 10 อย่าง คือขอบเขตทั้งหมดของระบบตัวอย่างวันนี้ — ชื่อบทบาทแอดมิน / พนักงานขาย / ผู้บริหาร เป็นตัวอย่าง ทีมของคุณกำหนดบทบาทเองได้ในข้อ 1.6



ผู้กระทำ	การกระทำ
ลูกค้าผู้สนใจ (ไม่ต้องล็อกอิน)	กรอกฟอร์มสนใจ
ทีมงานสมัครใหม่	สมัครใช้งาน · ล็อกอิน (เข้าได้หลังแอดมินอนุมัติ)
แอดมิน	อนุมัติผู้สมัคร · กำหนดบทบาท · มอบหมายลีด
พนักงานขาย	เปลี่ยนสถานะลีด · บันทึกการติดต่อ
ผู้บริหาร	ดู Dashboard · ดูรายงานยอดสมัคร

> **ทีมงาน กับ ลูกค้า ต่างกันอย่างไร**
> **ทีมงานสมัครใหม่** = พนักงานของสโมสรที่ขอบัญชีเข้า CRM ไม่ใช่ลูกค้า · **ลูกค้าผู้สนใจ** = คนนอกที่กรอกฟอร์มสนใจ กลายเป็นลีดในระบบ ไม่มีบัญชี

1.4 หน้าจอตามบทบาท

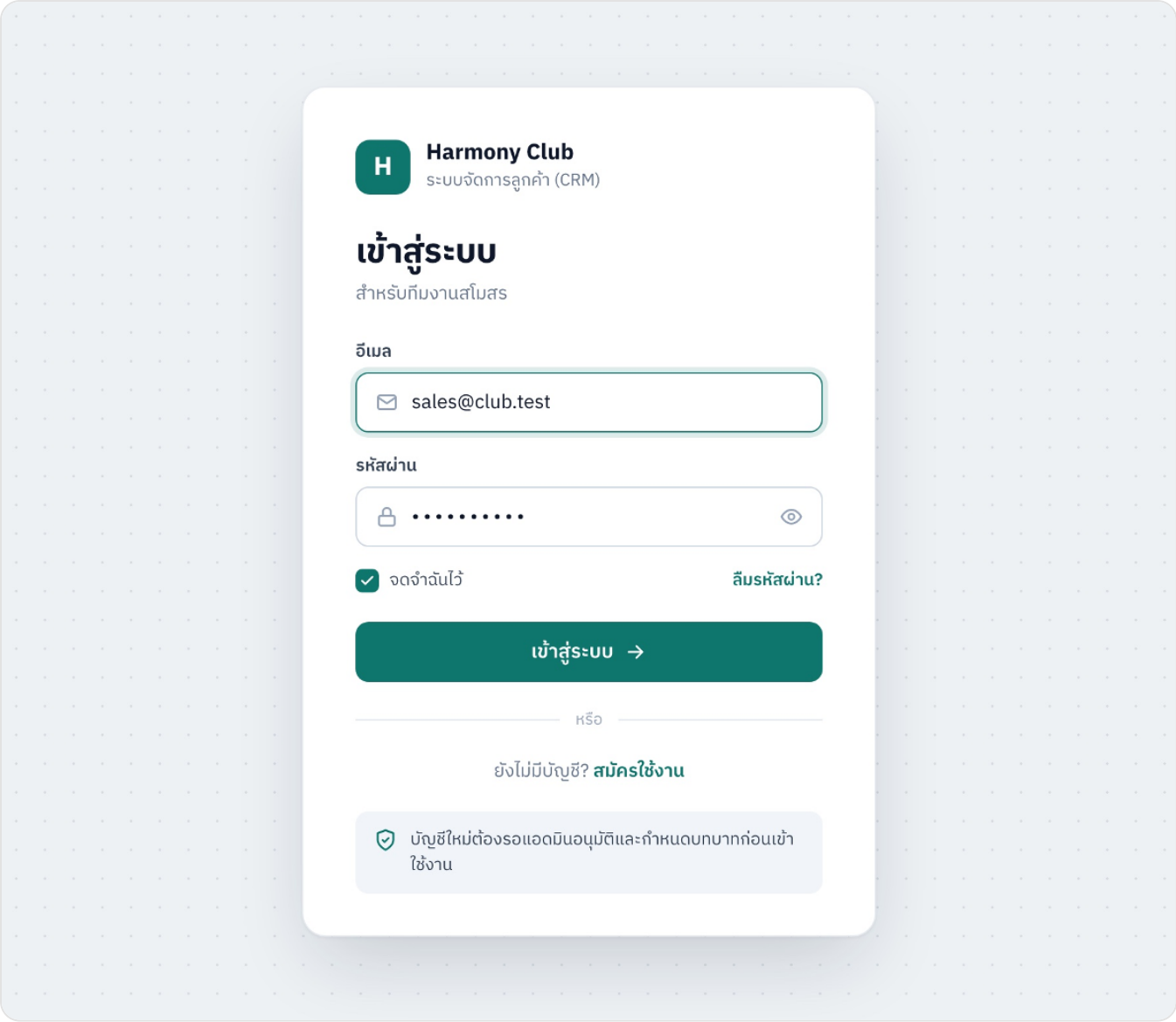
ดูหน้าจอตัวอย่าง (mockups/roles.html) ก่อนเขียนตารางสิทธิ์ — ระบบเดียวกัน ฐานข้อมูลเดียวกัน แต่เมนูและปุ่มที่แต่ละคนเห็นไม่เท่ากัน ขึ้นกับบทบาทที่ล็อกอินเข้ามา ภาพเหล่านี้ใช้แนบใน Prompt ตอนสร้างจริงด้วย · ชื่อบทบาทในภาพเป็นตัวอย่างของคลาส

4 ขั้นหลังกดล็อกอิน

01	02	03	04
กรอกอีเมล + รหัสผ่าน	ตรวจว่าเป็นใคร (Authentication)	อ่านบทบาท + สถานะจากตาราง profiles	เปิดเมนูตามสิทธิ์ (Authorization)

หน้าล็อกอิน

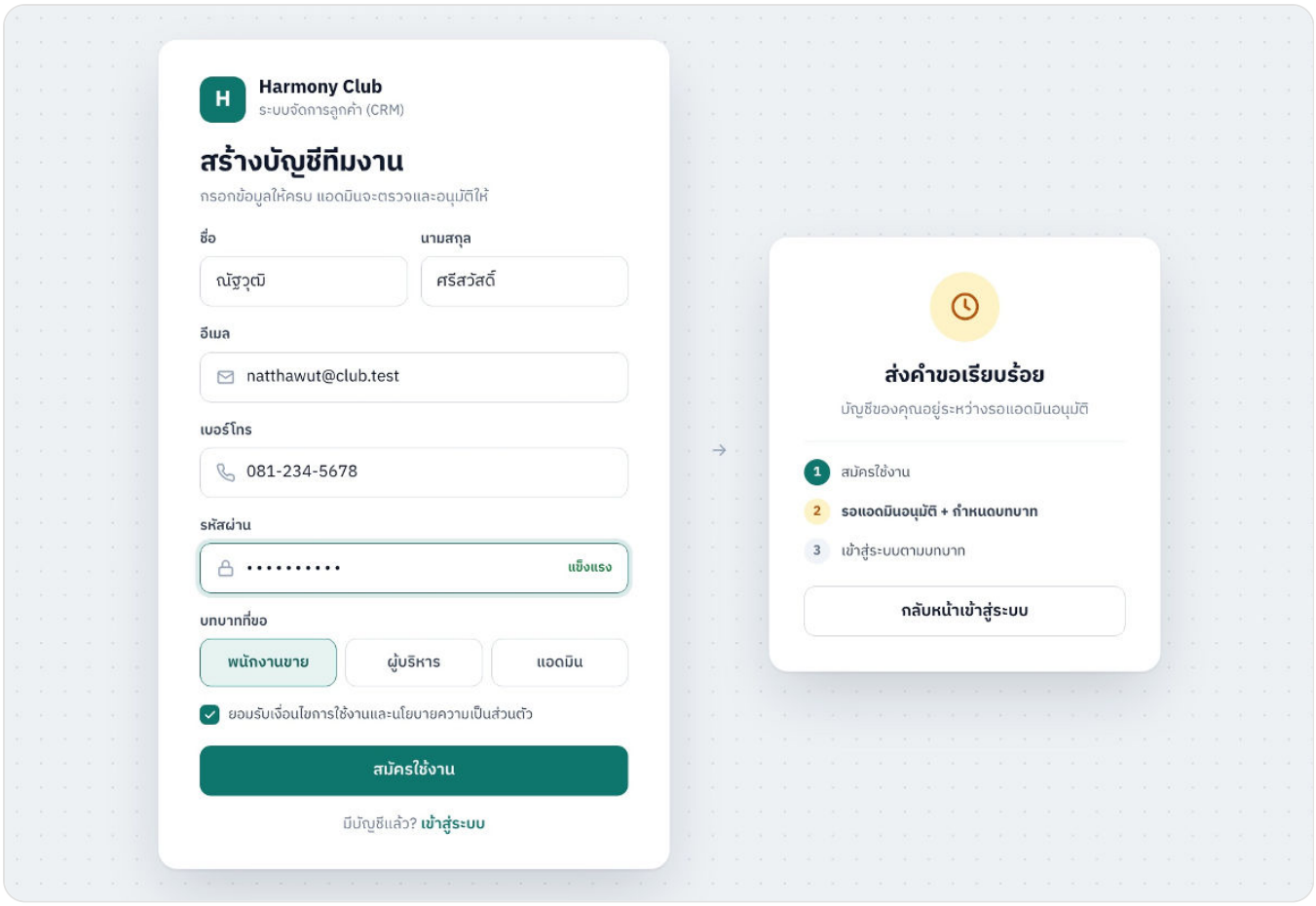
`/login · Phase 1`



การ์ดเดียว อีเมล + รหัสผ่าน · จดจำฉัน · ลืมรหัสผ่าน · ลิงก์สมัคร · หมายเหตุว่าบัญชีใหม่ต้องรออนุมัติ

หน้าสมัครใช้งาน

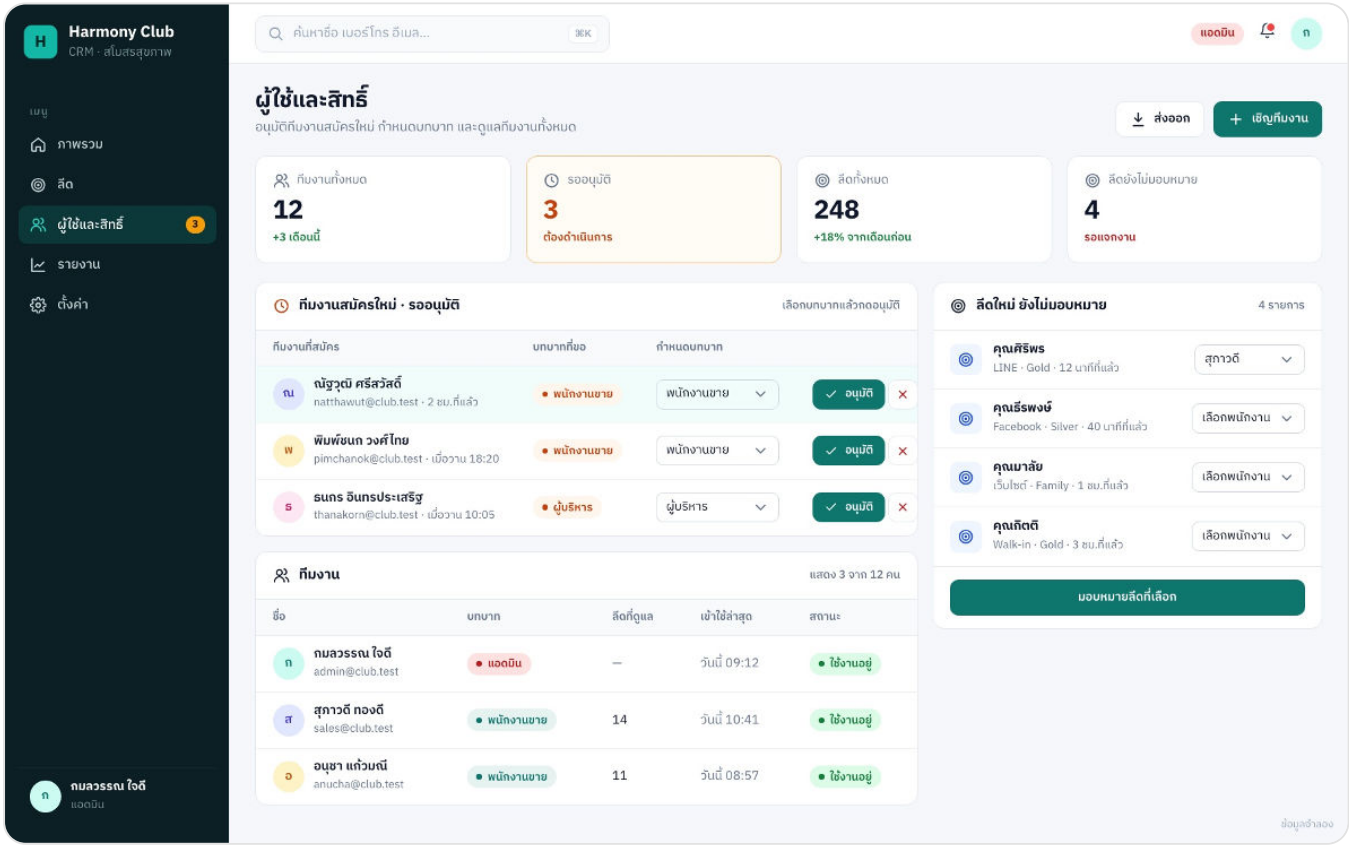
`/register · Phase 1`



ชื่อ นามสกุล อีเมล เบอร์โทร รหัสผ่าน บทบาทที่ขอ → หน้าจอ "ส่งคำขอเรียบร้อย รอแอดมินอนุมัติ"

แอดมิน · ผู้ใช้และสิทธิ์

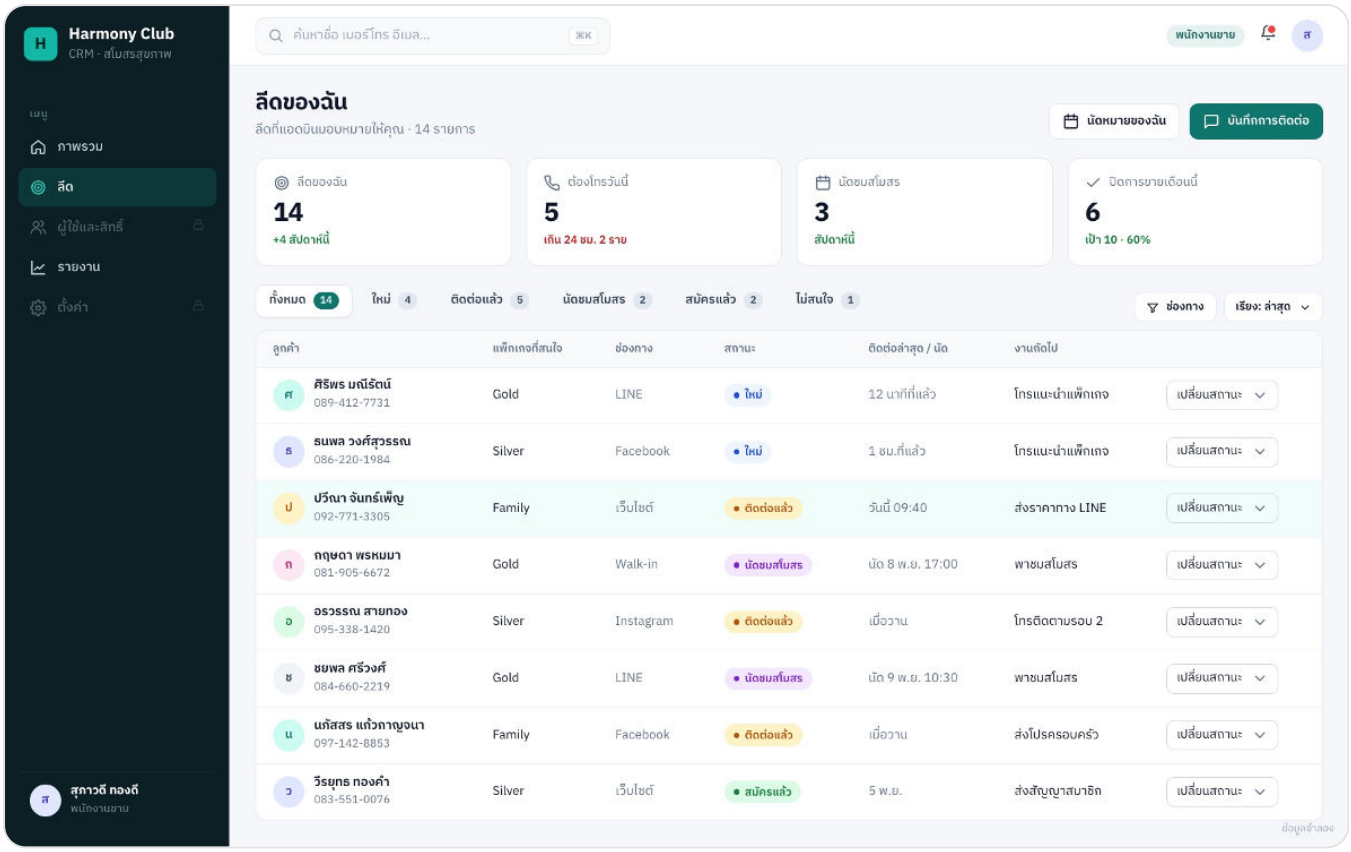
`role: admin · Phase 1`



ผู้สมัครรออนุมัติ (เลือกบทบาท + อนุมัติ/ปฏิเสธ) · ทีมงาน · ลีดใหม่ที่ยังไม่มอบหมาย

พนักงานขาย · ลีดของฉัน

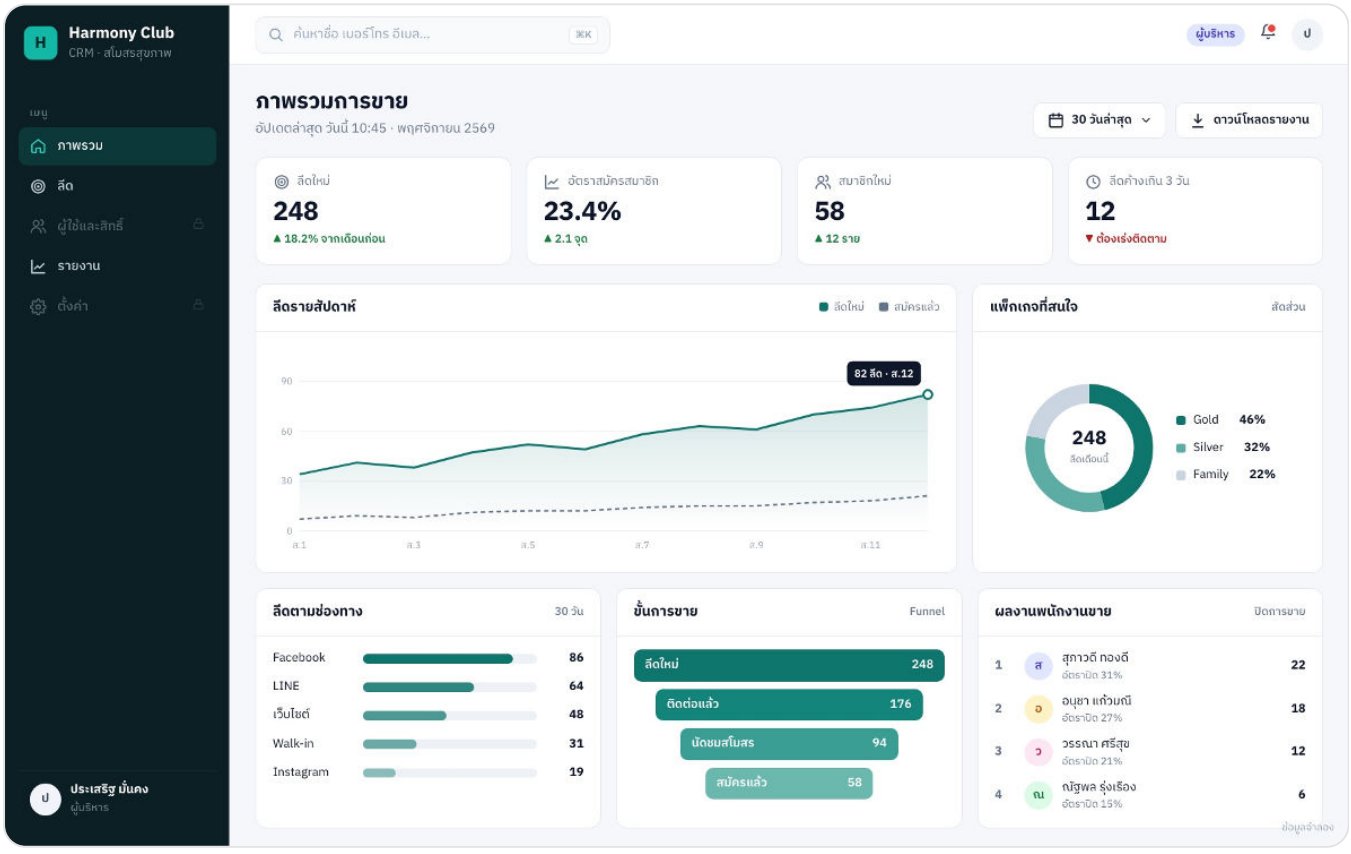
`role: sales · Phase 2`



ลีดของฉัน 14 รายการ แท็บตามสถานะ เปลี่ยนสถานะได้ · เมนูผู้ใช้/ตั้งค่าถูกล็อก

ผู้บริหาร · Dashboard

`role: manager · Phase 3`



KPI 4 ช่อง · กราฟลีดรายสัปดาห์ · แพ็กเกจที่สนใจ (โดนัท) · ลีดตามช่องทาง · ขั้นการขาย (Funnel) · ผลงานพนักงานขาย · ดูอย่างเดียว

1.5 Permission Matrix — ตารางสิทธิ์

การกระทำ	ทีมงานสมัครใหม่ (รออนุมัติ)	แอดมิน	พนักงานขาย	ผู้บริหาร
สมัครใช้งาน / ล็อกอิน	ได้	ได้	ได้	ได้
อนุมัติผู้สมัคร · กำหนดบทบาท	ไม่ได้	ได้	ไม่ได้	ไม่ได้
ดูรายชื่อลีด	ไม่ได้	ได้	เฉพาะของตัวเอง	ได้
มอบหมายลีด	ไม่ได้	ได้	ไม่ได้	ไม่ได้
เปลี่ยนสถานะ · บันทึกการติดต่อ	ไม่ได้	ได้	เฉพาะของตัวเอง	ไม่ได้
ลบลีด	ไม่ได้	ได้	ไม่ได้	ไม่ได้
ดู Dashboard	ไม่ได้	ได้	ไม่ได้	ได้

ลูกค้าผู้สนใจ (ไม่ล็อกอิน) กรอกฟอร์ม `/join` ได้อย่างเดียว · ตารางนี้เป็นตัวอย่างของคลาส — ใน CLAUDE.md ของคุณจะเป็นบทบาทและสิทธิ์ตามที่ทีมออกแบบ และใช้เป็นด่านตรวจตอนจบแต่ละเฟส

1.6 เขียน CLAUDE.md แบ่ง 3 เฟส

รวมทุกอย่างที่ออกแบบมา (1.3–1.5) ไว้ในไฟล์เดียว — Agent ถามเราทีละข้อไม่เกิน 6 ข้อ แล้วเขียนไฟล์ `CLAUDE.md` ที่ root ของโปรเจกต์ ยังไม่เขียนโค้ด

ก่อนส่ง เติมช่อง [ ] ใน Prompt เป็นระบบของทีมเอง

1. **บทบาทที่ 1–3:** ชื่อบทบาท · ทำได้ · ทำไม่ได้
2. **ทีมงานสมัครใหม่:** บทบาทไหนเป็นคนอนุมัติและกำหนดบทบาทให้
3. **ลูกค้าผู้สนใจ:** เก็บข้อมูลอะไรบ้าง เช่น ชื่อ เบอร์โทร อีเมล แพ็กเกจที่สนใจ ช่องทางที่รู้จักเรา

ยังไม่แน่ใจ ใช้ตัวอย่างของคลาส (แอดมิน / พนักงานขาย / ผู้บริหาร ตามข้อ 1.3–1.5) ได้เลย ข้อไหนเว้นไว้ Agent จะถามทีละข้อ

PROMPT DESIGN Workshop 1 · ออกแบบระบบ → เขียน CLAUDE.md แบ่ง 3 เฟส

```
เราจะสร้างระบบ CRM ของสโมสรสุขภาพ Harmony Club (ชื่อสมมติ · ใช้ข้อมูลจำลองเท่านั้น)
ช่วยฉันออกแบบระบบ แล้วเขียนเป็นไฟล์ CLAUDE.md ที่ root ของโปรเจกต์ (ถ้าใช้ Codex ให้ตั้งชื่อ AGENTS.md)

ระบบของเรา (เติมในวงเล็บเหลี่ยมให้ครบก่อนส่ง):
- บทบาทที่ 1: [ชื่อบทบาท เช่น แอดมิน]
  ทำได้: [...]
  ทำไม่ได้: [...]
- บทบาทที่ 2: [ชื่อบทบาท เช่น พนักงานขาย]
  ทำได้: [...]
  ทำไม่ได้: [...]
- บทบาทที่ 3: [ชื่อบทบาท เช่น ผู้บริหาร]
  ทำได้: [...]
  ทำไม่ได้: [...]
- ทีมงานสมัครใหม่: สมัครเองได้ แต่สถานะ "รออนุมัติ" จนบทบาท [ใคร] อนุมัติและกำหนดบทบาทให้
- ลูกค้าผู้สนใจ (ไม่ต้องล็อกอิน): กรอกฟอร์มสนใจสมัคร เก็บข้อมูล [เช่น ชื่อ เบอร์โทร อีเมล แพ็กเกจที่สนใจ ช่องทางที่รู้จักเรา]

ถ้าข้อไหนฉันยังเว้นไว้หรือไม่ชัด ให้ถามฉันทีละข้อก่อนเขียน (ไม่เกิน 6 ข้อ)

CLAUDE.md มี 6 หัวข้อ เขียนกระชับ ไม่เกิน 150 บรรทัด:
1. เป้าหมายระบบ (2-3 บรรทัด)
2. เครื่องมือ: Next.js (App Router · TypeScript · Tailwind) + Supabase (Auth + Database + Row Level
Security ผ่าน @supabase/ssr) + Git · ขึ้นเว็บบน Cloudflare ใน Phase 3
3. บทบาทและตารางสิทธิ์ (Permission Matrix เป็นตาราง) ตามที่ฉันเติมข้างบน
4. ตารางข้อมูล: profiles (ทีมงาน + บทบาท + สถานะอนุมัติ) · leads (ลูกค้าผู้สนใจ) · contact_logs (ประวัติการ
ติดต่อ)
5. กติกาทีม: คีย์อยู่ใน .env.local เท่านั้น · secret key ใช้ฝั่ง server เท่านั้น · ใช้ข้อมูลจำลอง · ทุกตารางเปิด Row
Level Security · ทำทีละเฟส เสร็จแล้วหยุดให้ฉันทดสอบ
6. แผนงาน 3 เฟส แต่ละเฟสมีเช็คลิสต์ "เสร็จเมื่อ…":
   - Phase 1 · สมัคร + ล็อกอิน + แบ่งสิทธิ์ตามบทบาท + อนุมัติทีมงานใหม่
   - Phase 2 · ลีด CRM: ฟอร์มสนใจ · มอบหมายลีด · แต่ละบทบาทเห็นลีดตามสิทธิ์
   - Phase 3 · Dashboard + ขึ้นเว็บ
ยังไม่ต้องเขียนโค้ด เขียนเสร็จแล้วสรุปให้ฉันตรวจ
```

ผลที่ได้ · โครงไฟล์ CLAUDE.md (ไม่เกิน 150 บรรทัด)

#	หัวข้อ	มีอะไร
1	**เป้าหมายระบบ**	2–3 บรรทัด
2	**เครื่องมือ**	Next.js (App Router · TypeScript · Tailwind) + Supabase (Auth + Database + Row Level Security ผ่าน @supabase/ssr) + Git · ขึ้นเว็บบน Cloudflare ใน Phase 3
3	**บทบาท + ตารางสิทธิ์**	Permission Matrix ตามบทบาทที่ทีมเติมใน Prompt (ตัวอย่าง: ข้อ 1.5)
4	**ตารางข้อมูล**	`profiles` · `leads` · `contact_logs` พร้อมชื่อคอลัมน์และความหมาย
5	**กติกาทีม**	คีย์อยู่ใน .env.local เท่านั้น · secret key ใช้ฝั่ง server เท่านั้น · ข้อมูลจำลอง · ทุกตารางเปิด Row Level Security · ทำทีละเฟส เสร็จแล้วหยุดให้ทดสอบ
6	**แผนงาน 3 เฟส**	แต่ละเฟสมีเช็คลิสต์ "เสร็จเมื่อ..."

Phase 1 · บ่ายนี้

สมัคร + ล็อกอิน + 3 บทบาท

อนุมัติทีมงานสมัครใหม่และกำหนดบทบาท ตามที่ออกแบบใน CLAUDE.md

Phase 2 · บ่ายนี้

ลีด CRM

ฟอร์มสนใจ · มอบหมายลีด · แต่ละบทบาทเห็นลีดตามสิทธิ์

Phase 3 · การบ้าน

Dashboard + ขึ้นเว็บ

หน้าภาพรวม · ขึ้นเว็บบน Cloudflare

ตรวจก่อนไปต่อ

อ่าน CLAUDE.md ที่ได้ทั้งไฟล์ บทบาทและตารางสิทธิ์ต้องตรงกับที่ทีมออกแบบไว้ ถ้าไม่ตรงให้สั่งแก้ตอนนี้ — แก้ในเอกสารถูกกว่าแก้ในโค้ดเสมอ

LUNCH BREAK

พักกลางวัน 12:30–13:30 น.



AFTERNOON · 13:30–16:30

ช่วงบ่าย · สร้างทีละเฟสจาก CLAUDE.md

บ่ายนี้ทำ Phase 1 และ Phase 2 ส่วน Phase 3 เป็นการบ้าน — ทุกเฟสวนรอบเดียวกัน 4 ขั้น



01 อ่าน CLAUDE.md แล้วทำ Phase N	02 วางคีย์ + SQL เอง	03 ทดสอบด่านบนจอจริง	04 ผ่านแล้วบันทึกด้วย Git

WORKSHOP 2 · 13:30–15:00

W2 Phase 1 · สมัคร + ล็อกอิน + 3 บทบาท

◎ **เป้าหมาย:** สมัครใช้งานได้ · บัญชีใหม่รออนุมัติ · ล็อกอินแล้วเห็นเมนูตามบทบาทที่ออกแบบใน CLAUDE.md และสิทธิ์ถูกล็อกถึงฐานข้อมูล

1. **คัดลอก PROMPT P1-BUILD**
   แนบหน้าจอตัวอย่าง 3 รูป (`mock-login` · `mock-register` · `mock-admin`)
2. **วาง Project URL + publishable key ใน `.env.local` เอง**
   ไม่ส่งคีย์ในแชต
3. **วาง `supabase/phase1.sql` ใน SQL Editor → Run**
4. **รันบนเครื่อง**
   `npm run dev` แล้วเปิด `http://localhost:3000`

5 **สมัครบัญชีตัวเอง**
แล้วรัน SQL 1 บรรทัดที่ Agent ให้ ตั้งเป็นบทบาทสูงสุด (ตัวอย่าง: แอดมิน)

6 **เพื่อนสมัคร 2 บัญชี**
แล้วอนุมัติ + กำหนดบทบาทตามที่ออกแบบไว้ (ตัวอย่าง: พนักงานขาย / ผู้บริหาร)

7 **ทดสอบด่าน 1 ตามเช็คลิสต์**
ผ่านแล้วบันทึกด้วย PROMPT GIT

PROMPT P1-BUILD — Phase 1 · สมัคร + ล็อกอิน + แบ่งสิทธิ์ตามบทบาท

```
อ่าน CLAUDE.md แล้วทำ Phase 1: สมัคร + ล็อกอิน + แบ่งสิทธิ์ตามบทบาทที่กำหนดใน CLAUDE.md

ฐานข้อมูล (เขียนไฟล์ supabase/phase1.sql ให้ฉันไปวางใน Supabase SQL Editor เอง)
- ตาราง profiles ผูกกับ auth.users: ชื่อ · อีเมล · เบอร์โทร · บทบาทที่ขอ · บทบาท (ตาม CLAUDE.md · ว่างได้) · สถานะ (pending / active / rejected)
- สมัครเสร็จ ให้ trigger สร้างแถว profiles อัตโนมัติ สถานะ pending
- Row Level Security ตามตารางสิทธิ์ใน CLAUDE.md: ทุกคนอ่านแถวของตัวเองได้ · เฉพาะบทบาทที่มีสิทธิ์อนุมัติเท่านั้นที่แก้บทบาท / สถานะได้ แม้เรียกฐานข้อมูลตรง

หน้าเว็บ (Next.js App Router + @supabase/ssr)
- /register สมัครใช้งาน: ชื่อ นามสกุล อีเมล เบอร์โทร รหัสผ่าน บทบาทที่ขอ → ขึ้น "รอการอนุมัติ"
- /login เข้าสู่ระบบ: ถ้ายังรออนุมัติ ให้แจ้งและห้ามเข้า
- /app หลังล็อกอิน แสดงเมนูตามบทบาทใน CLAUDE.md · หน้าอนุมัติทีมงานใหม่ (อนุมัติ / ปฏิเสธ + เลือกบทบาท) สำหรับบทบาทที่มีสิทธิ์ · หน้าของงานอื่นเว้นว่างไว้ทำใน Phase 2-3
- ดีไซน์สะอาด มืออาชีพ ตามหน้าจอตัวอย่างที่แนบ [แนบรูป mock-login · mock-register · mock-admin]

คีย์
- สร้างไฟล์ .env.local ที่มี NEXT_PUBLIC_SUPABASE_URL และ NEXT_PUBLIC_SUPABASE_PUBLISHABLE_KEY เว้นค่าว่างไว้ ฉันวางคีย์เอง
- ห้ามใช้ secret key ในโค้ดฝั่งหน้าเว็บ

ทำเสร็จแล้วบอก: วิธีรัน (npm run dev) · SQL 1 บรรทัดตั้งบัญชีแรกของฉันเป็นบทบาทที่มีสิทธิ์สูงสุด · เช็คลิสต์ทดสอบด่าน 1 แล้วหยุดรอฉันทดสอบ
```

ไฟล์	เก็บอะไร	ใครทำ
`.env.local`	`NEXT_PUBLIC_SUPABASE_URL` + `NEXT_PUBLIC_SUPABASE_PUBLISHABLE_KEY` (เปิดเผยได้) · Phase 2 เพิ่ม `SUPABASE_SECRET_KEY` (ฝั่ง server เท่านั้น)	Agent เว้นช่อง เราวางคีย์เอง · **ไฟล์ลับ ไม่ขึ้น Git**
`supabase/phase1.sql`	ตาราง profiles + trigger + Row Level Security	Agent เขียน เราวางใน SQL Editor

ด่าน 1 สมัครใหม่แล้วเข้าไม่ได้จนบทบาทที่มีสิทธิ์ (เช่น แอดมิน) อนุมัติ · สลับล็อกอิน 3 บทบาท เห็นเมนูต่างกัน · บทบาทที่ไม่มีสิทธิ์ (เช่น พนักงานขาย) แก้บทบาทตัวเองไม่ได้ แม้เรียกฐานข้อมูลตรง

GIT ผ่านด่านแล้ว บันทึกจุดย้อนกลับ

ใช้ Prompt นี้ทุกครั้งที่ผ่านด่าน — ตรวจไฟล์ลับ · commit · อัปเดตเช็คลิสต์ใน CLAUDE.md

PROMPT GIT

บันทึกจุดย้อนกลับ (ทุกครั้งที่ผ่านด่าน)

```
ผ่านด่านแล้ว ช่วยบันทึกจุดย้อนกลับด้วย Git:
- ตรวจก่อนว่าไม่มีไฟล์ลับ (.env.local) หรือคีย์ลับอยู่ในไฟล์ที่จะบันทึก
- commit พร้อมข้อความสั้น ๆ ว่าทำ Phase ไหนเสร็จ
- อัปเดตเช็คลิสต์ใน CLAUDE.md ว่าเฟสนี้เสร็จแล้ว
```

WORKSHOP 3 · 15:15–16:30

W3 Phase 2 · ลีด CRM

เป้าหมาย: ลูกค้ากรอกฟอร์มสนใจแล้วลีดเข้า CRM · มอบหมายลีด · แต่ละบทบาทเห็นและแก้ลีดตามสิทธิ์ใน CLAUDE.md (ตัวอย่าง: แอดมินมอบหมาย · พนักงานขายดูแลเฉพาะลีดของตัวเอง · ผู้บริหารดูอย่างเดียว)

ต่อจาก Phase 1 โดยไม่แก้ระบบล็อกอินที่ผ่านแล้ว ทำทีละชั้น เสร็จแต่ละชั้นหยุดทดสอบก่อนไปต่อ

ชั้น 1 · Database

`phase2.sql` : ตาราง leads + contact_logs + ลีดจำลอง 30 ราย + Row Level Security

ทำเอง: วางใน SQL Editor → Run

ชั้น 2 · Backend

Route Handler `app/api/join/route.ts` รับฟอร์มสนใจ สร้างลีดสถานะ new ด้วย secret key

ทำเอง: วาง `SUPABASE_SECRET_KEY` ใน `.env.local` แล้วรีสตาร์ต `npm run dev`

ชั้น 3 · Frontend

`/join` · หน้าลีดของแต่ละบทบาทตามตารางสิทธิ์ (ตัวอย่าง: ลีดทั้งหมด + มอบหมายของแอดมิน · ลีดของฉันของพนักงานขาย · ดูอย่างเดียวของผู้บริหาร)

สถานะลีด

new	contacted	tour	member	lost
ใหม่	ติดต่อแล้ว	นัดชมสโมสร	สมัครแล้ว	ไม่สนใจ

PROMPT P2-BUILD

Phase 2 · ลีด CRM

```
อ่าน CLAUDE.md แล้วทำ Phase 2: ลีด CRM (ต่อจาก Phase 1 · ห้ามแก้ระบบล็อกอินที่ผ่านแล้ว)
ทำทีละชั้น Database → Backend → Frontend เสร็จแต่ละชั้นหยุดให้ฉันทดสอบก่อน

1. Database (เขียน supabase/phase2.sql ให้ฉันไปวางเอง)
- leads: ข้อมูลลูกค้าตามที่กำหนดใน CLAUDE.md + status (new / contacted / tour / member / lost) +
assigned_to (→ profiles) + created_at / updated_at
- contact_logs: lead_id · staff_id · note · created_at
- ข้อมูลจำลอง 30 ลีด กระจายให้ทีมงานที่มีในระบบ
- Row Level Security ตามตารางสิทธิ์ใน CLAUDE.md: ใครเห็นลีดไหน · ใครแก้ได้ · ใครมอบหมายได้ · ใครลบได้

2. Backend
- Route Handler app/api/join/route.ts รับฟอร์มสนใจ ตรวจข้อมูล แล้วสร้างลีดสถานะ new ด้วย
SUPABASE_SECRET_KEY ใน .env.local (ฉันวางคีย์เอง · ใช้ฝั่ง server เท่านั้น)

3. Frontend
- /join ฟอร์มสนใจสมัครสำหรับลูกค้า (ไม่ต้องล็อกอิน)
- หน้าลีดของแต่ละบทบาทตามตารางสิทธิ์ใน CLAUDE.md: ค้นหา กรอง · มอบหมาย · แท็บตามสถานะ · เปลี่ยนสถานะ · บันทึกการ
ติดต่อ · ไม่แสดงปุ่มที่บทบาทนั้นไม่มีสิทธิ์
- ดีไซน์เดียวกับ Phase 1 ตามหน้าจอตัวอย่างที่แนบ [แนบรูป mock-sales]
```

ด่าน 2 ลูกค้ากรอก `/join` แล้วลีดเข้า · แอดมินมอบหมายให้พนักงานขาย A · A เห็นลีดนั้น พนักงานขาย B ไม่เห็น · A เปลี่ยนสถานะ + บันทึกการติดต่อได้ ลบไม่ได้ · ผู้บริหารดูได้ แก้ไม่ได้

ผ่านด่าน 2 แล้ว ใช้ PROMPT GIT เดิม (หน้า Phase 1) บันทึกจุดย้อนกลับอีกครั้ง

WRAP-UP · 16:30–17:00

สรุปวันนี้ + Phase 3 การบ้าน

SDLC + ระบบ 3 ชั้น

คนคิด ออกแบบ ตรวจ · Frontend · Backend · Database

Main Agent + CLAUDE.md

Main Agent ตัวเดียว สร้างทีละเฟสจาก CLAUDE.md ไฟล์เดียว

Authentication / Authorization

ล็อกสิทธิ์ถึงฐานข้อมูลด้วย Row Level Security

Phase 3 การบ้าน · Dashboard + ขึ้นเว็บ

1. หน้า Dashboard สำหรับบทบาทที่ดูภาพรวมได้ตาม CLAUDE.md (ตัวอย่าง: ผู้บริหาร) ตามหน้าจอตัวอย่าง `mock-manager` (แนบรูปใน Prompt)
2. ขึ้นเว็บบน Cloudflare ด้วย OpenNext adapter (`@opennextjs/cloudflare`) แล้ว `npm run deploy` · ตั้งคีย์ฝั่ง Cloudflare (ห้ามใส่ในโค้ด) · เพิ่ม URL เว็บใน Supabase → Authentication → URL Configuration
3. ทดสอบบนมือถือ ล็อกอินได้ทุกบทบาท และเห็นข้อมูลตรงตารางสิทธิ์

PROMPT P3-BUILD

Phase 3 · Dashboard + ขึ้นเว็บ (ทำต่อที่บ้าน)

```
อ่าน CLAUDE.md แล้วทำ Phase 3: Dashboard + ขึ้นเว็บ
1. หน้า Dashboard สำหรับบทบาทที่ดูภาพรวมได้ตาม CLAUDE.md ตามหน้าจอตัวอย่างที่แนบ [แนบรูป mock-manager]: ตัวเลขสรุป (ลีดใหม่ · อัตราสมัคร · สมาชิกใหม่ · ลีดค้าง) · กราฟลีดรายสัปดาห์ · ลีดตามช่องทาง · ขั้นการขาย · ผลงานทีม
2. ขึ้นเว็บบน Cloudflare: ตั้งค่าโปรเจกต์ Next.js ด้วย OpenNext adapter (@opennextjs/cloudflare) ให้ deploy ได้ด้วย npm run deploy · ตั้งคีย์ฝั่ง Cloudflare (ห้ามใส่ในโค้ด) · บอกให้ฉันเพิ่ม URL เว็บใน Supabase → Authentication → URL Configuration
3. บอกวิธีทดสอบบนมือถือ: ล็อกอินได้ทุกบทบาท และเห็นข้อมูลตรงตารางสิทธิ์
```

ด่าน 3 · การบ้าน เปิด URL เว็บจริงบนมือถือ → ล็อกอินได้ทุกบทบาท → บทบาทที่ดูภาพรวม (เช่น ผู้บริหาร) เห็น Dashboard · ผ่านแล้วใช้ PROMPT GIT บันทึกอีกครั้ง

NEXT STEPS

ต่อยอดด้วยวิธีเดียวกัน

เพิ่มเป็น Phase 4, 5 ... ใน CLAUDE.md แล้วสั่ง `อ่าน CLAUDE.md แล้วทำ Phase 4` — ทดสอบด่าน แล้วบันทึกด้วย Git เหมือนวันนี้

#	งานต่อยอด	ต่อจากของวันนี้อย่างไร
1	**Waiting List**	ตารางคิวรอสมัคร เชื่อมกับลีดและสมาชิก
2	**ระบบจอง**	ตารางการจองห้อง/บริการ เชื่อมกับสมาชิก
3	**ร่างสัญญา**	ดึงข้อมูลลีด/สมาชิกไปเติมแม่แบบ + ขั้นอนุมัติโดยคน
4	**Chatbot / LINE**	แชตตอบคำถาม แล้วส่งลีดเข้า CRM เดิม
5	**Agent ที่ทำงานในระบบ**	ลีดใหม่เข้า → AI สรุป/ร่างข้อความ → คนกดอนุมัติ

เช็คลิสต์ก่อนใช้ข้อมูลจริง

วันนี้ใช้ข้อมูลจำลองทั้งหมด ก่อนเปลี่ยนเป็นข้อมูลลูกค้าจริง ต้องผ่าน 4 ข้อนี้

- ☐ **ความปลอดภัย** — ทุกตารางเปิด Row Level Security ตรงตารางสิทธิ์ · คีย์ลับไม่อยู่ในโค้ดหน้าเว็บหรือใน Git
- ☐ **PDPA** — ขอความยินยอมก่อนเก็บและใช้ข้อมูล · มีนโยบายความเป็นส่วนตัว
- ☐ **สำรองข้อมูล** — ตั้งการสำรองฐานข้อมูลสม่ำเสมอ และลองกู้คืนจริง
- ☐ **รีเซ็ตรหัสผ่าน** — มีขั้นตอนลืมรหัสผ่านที่ใช้งานได้ · เปิดยืนยันอีเมลกลับ (คลาสปิดไว้)

TROUBLESHOOTING

แก้ปัญหาที่พบบ่อย

อาการ	สาเหตุ	ทางแก้
สมัครแล้วล็อกอินไม่ได้	ยังรออนุมัติ หรือยังไม่ปิด Confirm email	ให้แอดมินอนุมัติ · ปิด Confirm email ใน Supabase
ไม่มีใครเป็นแอดมิน	บัญชีแรกยังไม่ได้ตั้ง	รัน SQL 1 บรรทัดที่ Agent ให้
หน้าเว็บขึ้นว่าเชื่อมฐานข้อมูลไม่ได้	ยังไม่วางคีย์ใน `.env.local` หรือยังไม่รีสตาร์ต `npm run dev`	วาง Project URL + publishable key แล้วหยุด (Ctrl+C) และรัน `npm run dev` ใหม่
พนักงานขายเห็นลีดคนอื่น	Row Level Security ยังไม่ตรงตารางสิทธิ์	สั่ง "ตรวจ RLS ให้ตรงตารางสิทธิ์ใน CLAUDE.md"
Agent แก้หลายส่วนพร้อมกันจนงง	สั่งกว้างเกินไป	สั่งทีละเฟส ทีละชั้น หยุดให้ทดสอบ
เผลอ commit ไฟล์ลับ	`.gitignore` ไม่ครบ	ให้ Agent ลบออกจาก Git และเปลี่ยนคีย์ใหม่ทันที

QUICK REFERENCE

Prompt Cheat Sheet

Prompt ทั้งหมดในคู่มือ เรียงตามลำดับการใช้ — เปิดหน้าของขั้นนั้นแล้วกดคัดลอก

PROMPT	ขั้น	ใช้ทำอะไร
SETUP	เตรียม Env	ตรวจเครื่อง + สร้างโปรเจกต์ Next.js
DESIGN	Workshop 1	ออกแบบระบบ → เขียน CLAUDE.md แบ่ง 3 เฟส
P1-BUILD	Phase 1	สมัคร + ล็อกอิน + แบ่งสิทธิ์ตามบทบาท
GIT	ทุกด่าน	บันทึกจุดย้อนกลับ (ทุกครั้งที่ผ่านด่าน)
P2-BUILD	Phase 2	ลีด CRM
P3-BUILD	Phase 3	Dashboard + ขึ้นเว็บ (ทำต่อที่บ้าน)

ติดต่อสถาบัน ZenityX

โทร
062-262-9644

อีเมล
admin@zenityxai.com

ออกแบบหลักสูตรโดยสถาบัน ZenityX · Basic CRM System with Vibe Coding — ออกแบบระบบ CRM ให้ตอบโจทย์การใช้งานของคุณด้วย AI Agent · AI Agent Co-Working System · Day 1 · ข้อมูลทั้งหมดในคู่มือเป็นข้อมูลจำลองเพื่อการเรียนการสอน Harmony Club เป็นชื่อสมมติ