

สร้างเว็บด้วย ChatGPT (Codex) เชื่อมฐานข้อมูลจริงด้วย Supabase

เส้นทางจากแนวคิดสู่เว็บไซต์ที่เปิดใช้งานได้จริง ครอบคลุมขั้นตอนการพัฒนา
แบบ End-to-End สำหรับผู้เริ่มต้น

WORKSHOP GUIDE

คู่มือปฏิบัติการพัฒนาเว็บไซต์สมัยใหม่

31 กรกฎาคม 2026

Next.js • Tailwind • GitHub • Vercel

ภาพรวมหลักสูตรและผลลัพธ์การเรียนรู้

1

ครั้งที่ 1: วางรากฐาน UI และโครงสร้างเว็บ

2 ชั่วโมง 30 นาที

✓ ติดตั้ง Node.js, VS Code และ Codex extension

🛠️ พัฒนา UI ด้วย Next.js 16 App Router & Tailwind

✓ วิเคราะห์โจทย์และสร้างเอกสาร PRD.md

🏠 สร้าง Prototype และทดสอบ Flow หน้าเว็บเบื้องต้น

2

ครั้งที่ 2: เชื่อมต่อฐานข้อมูลและเผยแพร่จริง

2 ชั่วโมง 30 นาที

📄 ออกแบบ Schema ฐานข้อมูลใน database.md

🔒 จัดเก็บซอร์สโค้ดบน GitHub Repository

🔗 เชื่อมต่อ Supabase และจัดการ SQL CRUD

🚀 Deploy เว็บไซต์ผ่าน Vercel ให้พร้อมใช้งานจริง



เป้าหมายสุดท้าย (ULTIMATE GOAL)

เว็บไซต์ต้นแบบที่เชื่อมต่อข้อมูลจริง พร้อม URL สำหรับเผยแพร่ให้ผู้อื่นใช้งานได้ทันที

การเตรียมโจทย์และบทบาทผู้ใช้

กำหนดเป้าหมายโครงการ

ตัวอย่าง: ระบบบริหารจัดการลูกหนี้เงินยืมทรวง
ราชการ

- แก้ปัญหา (Problem): การติดตามหนี้สินมีความซับซ้อนและล่าช้า
- เป้าหมาย (Goal): สร้างระบบที่ช่วยให้ตรวจสอบสถานะได้แบบ Real-time
- ฟีเจอร์หลัก: ระบบยืมเงิน, ระบบคืนเงิน, ระบบรายงานสรุป

ทิศทางการออกแบบ

เน้นความสะดวกตา ใช้งานง่าย และสื่อถึงความเป็นมืออาชีพ



ส้มอ่อน



ขาว



เทาเข้ม

โทนสีหลัก: ส้มอ่อน (Primary) และ ขาว (Secondary)

บทบาทและสิทธิ์การเข้าถึง

บทบาท (Role)	สิทธิ์การเข้าถึง (Access Level)
ADMIN	เข้าถึงทุกหน้า จัดการข้อมูลผู้ใช้ และตั้งค่าระบบทั้งหมด
EXECUTIVE	ดูภาพรวมรายงานสรุป (Dashboard) และข้อมูลเชิงสถิติ
USER	สร้างรายการยืมเงิน และติดตามสถานะของตนเอง
STAFF	เห็นเฉพาะข้อมูลที่ได้รับมอบหมาย หรือข้อมูลของตนเองเท่านั้น
i การกำหนด Roles ที่ชัดเจนช่วยให้ AI ออกแบบ Database และ RLS ได้อย่างถูกต้อง	

การเตรียมสภาพแวดล้อมและเครื่องมือพัฒนา



ขั้นตอนการติดตั้งทีละขั้น

- ติดตั้ง Node.js**
ดาวน์โหลดและติดตั้งจาก nodejs.org
- ติดตั้ง VS Code**
ดาวน์โหลดและติดตั้งจาก code.visualstudio.com
- ติดตั้ง Codex Extension**
เปิด Extensions → ค้นหา Codex → Install → Trust Publisher
- สร้างไฟล์เดอร์โปรเจกต์**
สร้างบน Desktop แล้วเปิดด้วย VS Code

การตั้งค่าและเครื่องมือเสริม

- เปิดช่องแชท Codex ใน VS Code**
หลังติดตั้ง Codex แล้ว ให้เปิดแชทใน VS Code เพื่อเริ่มสั่งงาน AI ได้ทันทีภายใน Editor
[Codex Chat Panel](#)
- ติดตั้ง Skill Impeccable**
เครื่องมือช่วยออกแบบ UI ให้สวยงามและมีระบบ เหมาะสำหรับผู้เริ่มต้นที่ต้องการ UI ที่ดูเป็นมืออาชีพ
[UI Design Helper](#)
- โครงสร้างไฟล์เดอร์โปรเจกต์**
ตั้งชื่อไฟล์เดอร์ให้ชัดเจน เปิดไฟล์เดอร์ใน VS Code แล้วเริ่มต้นโปรเจกต์ Next.js จากภายใน Terminal
[Project Folder Setup](#)

การสร้าง PRD และโครงร่างหน้าเว็บต้นแบบ



1. สร้าง PRD.md

- ✓ ให้ AI วิเคราะห์ Requirement เบื้องต้น
- ✓ สรุบบทสรุปฟีเจอร์ลงในไฟล์เอกสาร
- ✓ กำหนด User Flow และหน้าเว็บทั้งหมด



2. เริ่มต้นโปรเจกต์

- 🏠 ใช้ Next.js 16 App Router
- 🛠️ ติดตั้ง TypeScript และ Tailwind CSS
- 🏗️ กำหนดโครงสร้างโฟลเดอร์ตามมาตรฐาน



3. พัฒนา UI First

- 🎨 สร้างหน้าจอตาม PRD (ยังไม่เชื่อมต่อ DB)
- 🛠️ ใช้ Skill Impeccable ช่วยออกแบบ
- 🛡️ เน้นความสวยงามและ User Experience



4. ตรวจสอบ Flow

- 👤 ทดสอบการกดปุ่มและการเปลี่ยนหน้า
- 🛠️ ตรวจสอบความถูกต้องกับ Requirement
- 🏠 ปรับปรุง UI ก่อนไปขั้นตอนฐานข้อมูล

Requirement > PRD.md > UI Prototype > Review & Refine

PRO TIP

รันโปรเจกต์เพื่อดูผลลัพธ์แบบ Real-time ผ่านเบราว์เซอร์ที่ `http://localhost:3000` หากเปิดไม่ได้ให้ขอ AI ช่วยรันคำสั่ง `npm run dev`



การออกแบบฐานข้อมูลจากข้อมูลจริงและ Excel

กระบวนการออกแบบ

- 1 วิเคราะห์ Excel & Features
นำไฟล์ข้อมูลจริงเข้าโปรเจกต์เพื่อให้ AI วิเคราะห์ฟีเจอร์ที่จำเป็น
- 2 สร้างเอกสาร database.md
สรุปตาราง (Tables), ฟیلด์ (Fields) และความสัมพันธ์ (Relationships)
- 3 ตรวจสอบความสอดคล้อง
รีวิวข้อมูล Excel ฐานข้อมูล และหน้า UI ให้ตรงกัน 100%

💡 **Pro-Tip:** หากต้องการแก้ไขโครงสร้างให้สั่ง AI ด้วยภาษาธรรมชาติ เช่น "เพิ่มฟิลด์ที่อยู่และเบอร์โทรในตาราง user"

ของคุณต้น

ให้นำไฟล์ excel มาใส่ในไฟล์เตอร์โปรเจกต์นี้ก่อน แล้วจึงใส่ prompt นี้

Prompt: วิเคราะห์ฟีเจอร์และ Requirement ทั้งหมดของเว็บไซต์ โดยใช้ไฟล์ Excel ชื่อ <ทะเบียนลูกหนี้

69 ใช้ปกติ.1> เป็นข้อมูลอ้างอิงหลัก ให้ทำงานตามลำดับนี้:

1. วิเคราะห์ข้อมูลจากไฟล์ Excel และฟีเจอร์ของเว็บไซต์
2. ออกแบบฐานข้อมูลให้รองรับระบบจริง
3. สร้างไฟล์ database.md เพื่อสรุปโครงสร้างฐานข้อมูล ได้แก่ ตาราง ฟیلด์ ชนิดข้อมูล และความสัมพันธ์
4. ตรวจสอบว่าข้อมูลใน Excel สอดคล้องกับฐานข้อมูลและหน้าเว็บหรือไม่
5. หากข้อมูลไม่ตรงหรือไม่เพียงพอ ให้เสนอรายการข้อมูลที่ต้องแก้ไข เพิ่มเติม หรือตัดออก
6. หลังจากออกแบบฐานข้อมูลและสร้าง database.md เสร็จแล้ว ค่อยแก้ไขหน้าเว็บให้สอดคล้องกับ

ข้อมูลจริงและโครงสร้างฐานข้อมูลใหม่

หมายเหตุ: ให้ยึดข้อมูลเดิมใน Excel และข้อมูลที่เกี่ยวข้องเป็นหลัก

2. หลังจากที่ได้เอกสาร database.md มาแล้ว ให้เราตรวจสอบ และอ่านคร่าวๆตรวจสอบว่าตรงกับสิ่งที่เราต้องการไหม หากเราโอเคแล้วไปที่ ข้อ 3 ได้
 - หากยังไม่โอเค อยากรู้แก้ไข ให้พิมพ์สิ่งที่ต้องการแก้ไขลงไป เช่น
ต้องการเก็บข้อมูลของ user เพิ่มเติม เก็บเบอร์โทรศัพท์ ที่อยู่ เพิ่มเติมด้วย
แล้วส่งให้กับ codex ในแชทได้เลย
3. หากตรวจสอบเอกสาร database.md แล้ว เราจะเริ่มสร้างฐานข้อมูลกัน โดยให้นำหน้าเว็บเรา เชื่อมต่อ

กับ supabase เพื่อให้สามารถ เพิ่ม ลบ แก้ไข ข้อมูลได้จริง

โดยนำ prompt ด้านล่าง ไปใส่ในช่องแชท codex ได้เลย

รายการตรวจสอบ (Validation)

- ✓ ฟیلด์ข้อมูลใน database.md ครบถ้วนตาม Excel
- ✓ ชนิดข้อมูล (Data Types) ถูกต้องและเหมาะสม
- ♥ หน้า UI แสดงผลข้อมูลได้ตรงตามโครงสร้างฐานข้อมูล

การเชื่อมต่อ Supabase และความปลอดภัยของข้อมูล

🔌 ขั้นตอนการเชื่อมต่อ

1 สร้าง Supabase Project
เข้า supabase.com → สร้างโปรเจกต์ใหม่ → ตั้งชื่อและรหัสผ่านฐานข้อมูล

2 รับ Project URL และ API Keys
ไปที่ Settings → API → คัดลอก Project URL, Anon Key และ Service Role Key

3 ตั้งค่า Environment Variables
สร้างไฟล์ .env.local ในโปรเจกต์ → วาง Key ตามตัวแปรที่กำหนด

4 ให้ Codex เชื่อมตาม database.md
สั่ง AI เชื่อม Supabase client ตามโครงสร้างที่ออกแบบไว้

5 รับ SQL และทดสอบ
เปิด SQL Editor ใน Supabase → วาง .sql → Run → ตรวจสอบตารางและทดสอบ CRUD

🔑 ประเภท API KEY

- Anon / Public Key** Client-Side
- ใช้ฝั่ง Client ได้อย่างปลอดภัย
 - ต้องเปิด Row Level Security (RLS) ให้เหมาะสม
 - สามารถเปิดเผยใน Frontend Code ได้

- Service Role Key** Server-Only
- เปรียบเสมือน "รหัสผ่านหลัก" ของฐานข้อมูล
 - ใช้เฉพาะฝั่ง Server เท่านั้น
 - ห้ามเปิดเผยใน Frontend, สไลด์ หรือแชท

.ENV.LOCAL

```
NEXT_PUBLIC_SUPABASE_URL=https://xxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJ...
SUPABASE_SERVICE_ROLE_KEY=eyJ...
```

🛡️ ข้อควรระวังด้านความปลอดภัย

⚠️ API Key คือรหัสผ่าน ต้องปกป้องเสมอ
ห้ามแชร์ Service Role Key ในช่องแชท, สไลด์, screenshot หรือ repository ที่เป็น Public โดยเด็ดขาด

🚫 ห้าม Commit ไฟล์ .env
เพิ่ม .env.local ใน .gitignore ทุกครั้งก่อน push ขึ้น GitHub

🚫 ห้ามวาง Key ใน Frontend Code
Service Role Key ต้องอยู่ใน Server-side functions เท่านั้น เช่น API Routes หรือ Server Actions

✅ ใช้ Environment Variables uu Vercel
ตั้งค่า Env Vars ใน Vercel Dashboard แทนการฝัง Key ในโค้ด

🛡️ Row Level Security (RLS)
เปิด RLS บนทุกตารางเพื่อควบคุมว่าผู้ใช้แต่ละ Role เห็นข้อมูลใดได้บ้าง ป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

การจัดการ SQL และการแก้ไขข้อผิดพลาด

ขั้นตอนการรับ SQL บน Supabase



1. รับไฟล์ .sql จาก Codex

สั่ง AI ให้สร้าง SQL Schema ตาม database.md เพื่อเตรียมโครงสร้างตาราง



2. เปิด SQL Editor

เข้าสู่หน้า Dashboard ของ Supabase และเลือกเมนู SQL Editor สำหรับจัดการฐานข้อมูล



3. วางโค้ดและ Run

คัดลอกคำสั่ง SQL ทั้งหมดมาวางใน Editor แล้วกด Run จากนั้นตรวจสอบความสำเร็จ



เทคนิคการแก้ปัญหาด้วย AI



เมื่อพบข้อผิดพลาด (Error Handling)

- ✓ คัดลอกข้อความ Error ทั้งหมด: ยื่อนำไปเฉพาะส่วน ให้ AI วิเคราะห์จาก Log จริง
- ✓ วางในแชท Codex: แจ้ง AI ว่าเกิด Error ระหว่างการรัน SQL บน Supabase
- ✓ แทนที่และทดสอบ: นำ SQL ฉบับที่ AI แก้ไขแล้วไปวางทับใน Editor และรันใหม่



Copy Error



Ask AI



Fix SQL



Success



ทดสอบ CRUD ซ้ำทุกหน้า

หลังแก้ไข SQL สำเร็จให้กลับไปทดสอบ เพิ่ม/ลบ/แก้ไข ข้อมูลจริงใน localhost:3000

การจัดเก็บซอร์สโค้ดบน GitHub



1. สร้าง Repository

สร้างพื้นที่เก็บโปรเจกต์ใหม่บน GitHub กำหนดชื่อและเลือกระดับการเข้าถึง (Public/Private)



2. เชื่อมต่อ VS Code

ใช้ AI ช่วย Login และ Authorize สิทธิ์การใช้งาน Git/GitHub ภายในตัวโปรแกรม



3. Commit & Push

บันทึกการเปลี่ยนแปลงของโค้ด และส่งขึ้นไปยัง GitHub Repository URL ที่กำหนด

supabase ได้

กรณีหากใส่ไฟล์ sql ที่ supabase (sql editor) แล้วเกิด error

1. Copy Error ให้หมด
2. นำมาวางที่ช่องแชท codex กด enter
3. รอ ai ทำงานเสร็จ
4. Copy โค้ดในไฟล์ sql ทั้งหมดอีกครั้ง
5. ลบโค้ด sql เก่าออก นำไปวางใน supabase → sql editor → กดปุ่ม run
6. รอบนี้ควรจะขึ้น success
7. ลองเล่นเว็บของตัวเอง localhost:3000

6. การนำโค้ดเก็บบน github

1. เข้า github <https://github.com/>

- กดปุ่ม new เพื่อสร้างโปรเจกต์ 
- กรอกชื่อโปรเจกต์ของตัวเอง ในช่อง Repository name

Owner *	Repository name *
 nakdev-23	



ข้อควรระวัง: ห้าม Commit ข้อมูลความลับ!

ตรวจสอบ .gitignore เพื่อไม่ให้ไฟล์ .env หรือ API Keys รั่วไหลสู่สาธารณะ

No .env

No Secrets

Use .gitignore

การเผยแพร่เว็บไซต์ด้วย Vercel

1 เชื่อมต่อ GitHub

เข้าสู่ระบบ Vercel และเลือก Repository ที่ต้องการนำขึ้นใช้งาน

2 ตั้งค่า Environment Variables

ระบุ Project URL และ API Keys จาก Supabase ให้ถูกต้องเพื่อความปลอดภัย

3 Deploy & Production

กด Deploy เพื่อรับโดเมนเนม (URL) สำหรับส่งให้ผู้อื่นเข้าใช้งานได้จริง

🔔 ระบบอัปเดตอัตโนมัติ (CI/CD)



แก้ไขโค้ด



Push GitHub

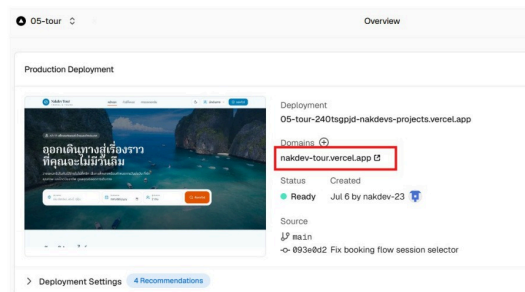


Auto Deploy



Live Site

- กดปุ่ม Continue to dashboard
- Deploy หรือนำขึ้นระบบเสร็จเรียบร้อย
- ในหน้า dashboard จะมีคำว่า domain และมีลิงค์อยู่ด้านล่าง นั่นคือลิงค์ของเรา สามารถนำไปใช้งานได้เลย หรือส่งต่อให้คนอื่นใช้งานได้เลย



หากในอนาคตมีการแก้ไข

- หากมีไฟล์ .sql เพิ่มเติมให้นำไปใส่ใน supabase → sql editor → ใส่ในช่องว่าง → run หากโอเคแล้วจะนำโค้ดขึ้นระบบ
- บอก AI (VS Code) ให้นำโค้ดขึ้น github ให้หน่อย
- หลังจากที่โค้ดถูกนำขึ้น github เรียบร้อยแล้ว มันจะ deploy หรือนำขึ้นระบบที่ vercel ให้อัตโนมัติ
- รอประมาณ 2-3 นาที ให้ลองเล่นที่เว็บตัวเองใหม่อีกครั้ง



Deployment Successful

สถานะการทำงานจริงบน Cloud พร้อมระบบ Auto-Deployment

การตรวจสอบคุณภาพและเกณฑ์ความสำเร็จของงาน

Definition of Done

โปรเจกต์จะสมบูรณ์ได้ เมื่อผ่าน 5 เกณฑ์หลัก

- ✓ หน้าเว็บและ User Flow ครบถ้วนตาม PRD
- ✓ สิทธิ์การเข้าถึง (Roles/RLS) ทำงานถูกต้อง
- ♥ ระบบฐานข้อมูล (CRUD) เชื่อมต่อข้อมูลจริง
- 🏠 ไม่มี Secrets/API Key รั่วไหลในโค้ด
- 🚀 Deploy บน Vercel และใช้งานได้จริงผ่านลิงก์

QUALITY ASSURANCE PROCESS



ตรวจสอบมาตรฐานโค้ด (Clean Code)

ตรวจสอบการแยก Routes, Components, และ Logic ให้เป็นระเบียบตามมาตรฐาน Next.js

Prompt: "Review code structure & clean up"



วิเคราะห์ฟังก์ชันที่ขาดหาย

เปรียบเทียบ UI ปัจจุบันกับ PRD เพื่อหาหน้าหรือ Modal ที่ยังไม่ได้สร้างให้ครบ Flow

Prompt: "Analyze missing features from PRD"



ความปลอดภัยและการตั้งค่า

ตรวจสอบไฟล์ .gitignore และ Environment Variables เพื่อป้องกันข้อมูลรั่วไหล

Prompt: "Security & Env Vars audit"