vibe code

เช็กลิสต์ก่อนปล่อยแอป

ติ๊กให้ครบก่อนเอาแอปที่ให้ AI ช่วยเขียนขึ้นใช้งานจริง
พร้อมพรอมต์ให้ AI ช่วยตรวจความปลอดภัยให้

เช็กลิสต์ (1/2)

1 คีย์และรหัสลับ

ปี 2024 มีรหัสลับหลุดบน GitHub สาธารณะ 23.8 ล้านครั้ง (GitGuardian)

- [ ] ไม่มี API key / รหัสผ่าน อยู่ในโค้ดฝั่งหน้าเว็บ (คนเปิดดูได้หมด)
- [ ] เก็บความลับไว้ใน `.env` และมี `.env` อยู่ใน `.gitignore`
- [ ] ถ้าเคย commit key ขึ้นไปแล้ว: **สร้าง key ใหม่** (ลบไฟล์อย่างเดียวไม่พอ ยังอยู่ในประวัติ Git)
- [ ] เปิดระบบกัน push ความลับของ GitHub (secret scanning / push protection)
- [ ] ตั้งวงเงินหรือแจ้งเตือนค่าใช้จ่ายของ API ที่ใช้

2 สิทธิ์เข้าถึงข้อมูล

มีการสแกนแอปที่สร้างด้วย AI 1,645 แอป เจอ 170 แอปเปิดให้อ่านข้อมูลได้โดยไม่ต้องล็อกอิน

- [ ] ทุกตารางในฐานข้อมูลเปิดการล็อกสิทธิ์แล้ว (ถ้าใช้ Supabase = เปิด RLS + เขียน policy)
- [ ] ทุก API เช็กว่า "คนที่ขอ เป็นเจ้าของข้อมูลนี้จริงไหม" ไม่ใช่แค่เช็กว่าล็อกอินแล้ว
- [ ] ลองทดสอบเอง: ล็อกอินอีกบัญชี หรือเปิดแบบไม่ล็อกอิน แล้วลองดึงข้อมูลของคนอื่น
- [ ] ส่งข้อมูลไปหน้าเว็บเฉพาะที่จำเป็น (ไม่ส่งทั้งแถวที่มีข้อมูลส่วนตัว)

3 แพ็กเกจที่ติดตั้ง

งานวิจัย USENIX Security 2025 พบโค้ดจาก AI ราว 1 ใน 5 มีชื่อแพ็กเกจที่ไม่มีอยู่จริง

- [ ] เช็กชื่อแพ็กเกจในเว็บทางการ (npmjs.com / pypi.org) ก่อนติดตั้งทุกครั้ง
- [ ] ดูยอดดาวน์โหลด วันที่สร้าง และผู้ดูแล ถ้าเพิ่งสร้างไม่นานแต่ชื่อคล้ายตัวดัง ให้ระวัง
- [ ] commit ไฟล์ lockfile (เช่น `package-lock.json`) และรัน `npm audit` เป็นระยะ

เช็กลิสต์ (2/2)

4 Git และการสำรองข้อมูล

ปี 2025 มีเคส AI ลบฐานข้อมูลระหว่างทดลองเขียนแอป ทั้งที่สั่งห้ามแก้ไว้ (กู้คืนได้เพราะมี backup)

- [ ] ใช้ Git ตั้งแต่วันแรก และ commit ทุกครั้งที่ได้ผลลัพธ์ที่ใช้งานได้
- [ ] ฐานข้อมูลมี backup อัตโนมัติ และ**เคยลองกู้คืนจริง**อย่างน้อย 1 ครั้ง
- [ ] แยกฐานข้อมูลจริง (production) กับตัวทดสอบ (development) ออกจากกัน
- [ ] AI ที่ช่วยเขียนโค้ด ไม่มีสิทธิ์เข้าฐานข้อมูลจริงโดยตรง

5 เข้าใจโค้ดของตัวเอง

- [ ] ให้ AI วางแผนก่อนลงมือ (โหมด plan) แล้วค่อยอนุมัติ
- [ ] ให้ AI อธิบายทุกครั้งว่าแก้อะไร ไฟล์ไหน เพราะอะไร
- [ ] อ่านคำสั่งที่ AI จะรันก่อนกดอนุญาต โดยเฉพาะคำสั่งลบ (`rm`, `DROP`, `DELETE`)
- [ ] มีเทสต์ครอบส่วนสำคัญ: ล็อกอิน การชำระเงิน การลบข้อมูล

+ ก่อนขึ้นใช้งานจริง

- [ ] เว็บใช้ HTTPS ทุกหน้า
- [ ] จำกัดจำนวนครั้ง (rate limit) ที่หน้า login และ API สำคัญ
- [ ] ตรวจข้อมูลที่ผู้ใช้กรอกทุกช่อง ทั้งฝั่งหน้าเว็บและฝั่งเซิร์ฟเวอร์
- [ ] ข้อความ error ไม่โชว์รายละเอียดภายในระบบ (ชื่อตาราง, stack trace)
- [ ] รันพรอมต์ตรวจความปลอดภัยในหน้าถัดไปแล้ว

พรอมต์ให้ AI ตรวจความปลอดภัย

ก๊อปไปวางใน AI ที่เปิดโปรเจกต์ของคุณอยู่ (เช่น Claude Code, OpenCode, Cursor) ใช้ได้ทุกตัว

พรอมต์ 1: ตรวจทั้งโปรเจกต์

```
ช่วยตรวจความปลอดภัยของโปรเจกต์นี้ก่อนขึ้นใช้งานจริง
ห้ามแก้ไฟล์ จนกว่าฉันจะอนุมัติ ให้รายงานอย่างเดียวก่อน

ตรวจหัวข้อเหล่านี้:
1. ความลับ: API key / รหัสผ่าน / token ที่อยู่ในโค้ดหรือหลุดไปฝั่งหน้าเว็บ
   และเช็กว่า .env อยู่ใน .gitignore
2. สิทธิ์: ทุก API และทุกตารางในฐานข้อมูล เช็กว่าผู้ใช้เข้าถึง
   ได้เฉพาะข้อมูลของตัวเองจริงไหม (รวม RLS ถ้ามี)
3. แพ็กเกจ: ทุกตัวใน dependencies มีอยู่จริงและเป็นตัวทางการไหม
   ตัวไหนน่าสงสัย หรือมีช่องโหว่ที่รู้แล้ว
4. ข้อมูลที่ผู้ใช้กรอก: จุดเสี่ยง injection / XSS / อัปโหลดไฟล์
5. คำสั่งอันตราย: โค้ดที่ลบหรือเขียนทับข้อมูลจำนวนมาก
6. การตั้งค่า: CORS, rate limit, ข้อความ error ที่เผยข้อมูลภายใน

รายงานเป็นตาราง: ระดับ (สูง/กลาง/ต่ำ) | ไฟล์:บรรทัด |
ปัญหา | วิธีแก้ เรียงจากระดับสูงไปต่ำ
```

พรอมต์ 2: กฎประจำโปรเจกต์ (วางใน AGENTS.md / CLAUDE.md)

```
กฎการทำงานในโปรเจกต์นี้:
- วางแผนและอธิบายก่อนแก้ทุกครั้ง แล้วรอฉันอนุมัติ
- ห้ามรันคำสั่งที่ลบข้อมูล (rm -rf, DROP, DELETE ทั้งตาราง)
  โดยไม่ถามก่อน
- ห้ามแตะฐานข้อมูล production
- ห้ามใส่ความลับในโค้ด ให้อ่านจาก environment variable
- ก่อนติดตั้งแพ็กเกจใหม่ ให้บอกชื่อและเหตุผล และยืนยันว่ามีจริง
- แก้เสร็จแล้วรันเทสต์ และสรุปว่าแก้ไฟล์ไหนบ้าง
```

หมายเหตุ: AI ช่วยตรวจได้ดี แต่ไม่ได้รับประกันว่าจะเจอทุกจุด แอปที่เก็บข้อมูลส่วนตัวหรือรับเงินจริง ควรให้คนที่เชี่ยวชาญด้านความปลอดภัยช่วยดูอีกชั้น

ที่มาของตัวเลขในคลิป

- **รหัสลับหลุด 23.8 ล้านครั้ง (ปี 2024):** GitGuardian, The State of Secrets Sprawl 2025 (blog.gitguardian.com)
- **170 จาก 1,645 แอปเปิดข้อมูล:** การสแกนแอปที่สร้างด้วย AI ปี 2025 ภายหลังบันทึกเป็น CVE-2025-48757 (ผู้ให้บริการแย้งว่าเป็นการตั้งค่าของผู้ใช้แต่ละราย)
- **แพ็กเกจที่ไม่มีจริง ~1 ใน 5:** งานวิจัย "We Have a Package for You!" USENIX Security 2025 (ตัวเลขอ้างอิงจากบทสรุปของงานวิจัย)
- **AI ลบฐานข้อมูล:** เหตุการณ์ระหว่างทดลอง vibe coding บน Replit เดือน ก.ค. 2025 รายงานโดย OECD AI Incidents, eWeek, heise

มีหลุมไหนที่เคยตกมาแล้ว? เล่าให้ฟังหน่อย

ติดตาม **keaykub.dev** · เทคยากๆ เล่าให้ง่ายใน 1 นาที

เอกสารนี้แจกฟรี แชร์ต่อได้ ข้อมูล ณ ต.ค. 2026