จากใจผู้เขียน

เรื่องทั้งหมดมันเริ่มจาก มือถือลูกรักที่ใช้มาเกือบ 7 ปีของผมครับ

ย้อนไปประมาณ 2 ปีที่แล้ว ผมใช้มือถือ Samsung Galaxy และใช้แอป Samsung Notes ที่ติดมากับเครื่องเป็นประจำ ซึ่งต้องบอกว่าใช้ดีมากครับ จดอะไรก็สะดวก มีระบบค้นหาบันทึก ล็อกโน้ตที่ไม่อยากให้คนอื่นเปิดดูได้ แถมยังสำรองข้อมูลไว้บน Cloud อีกด้วย ไม่ต้องคอยกังวลว่าข้อมูลจะหาย

ผมใช้เก็บโน้ตสารพัดอย่างเลยครับ จนวันหนึ่งมือถือเสีย!

เอาละสิ จะซื้อเครื่องใหม่ดีไหม หรือถือโอกาสเปลี่ยนไปใช้ iPhone เลยดีนะ?

แต่พอเริ่มดูเรื่องการย้ายข้อมูล กลับเจอปัญหาที่ไม่เคยคิดถึงมาก่อนครับ

แอปที่ผมใช้จดโน้ตมาตลอด ใช้ได้เฉพาะในระบบของ Samsung!

จะเปิดใช้งานผ่านเว็บออนไลน์จากเครื่องไหนก็ไม่ได้ ส่วนการใช้บนคอมพิวเตอร์ก็ต้องซิงก์ฐานข้อมูลลงมาด้วย ซึ่งข้อมูลของผมมีอยู่หลายกิกะไบต์แน่นอนครับ

ผมเลยเริ่มมองหาแอปอื่นที่ใช้ได้ทั้งบนมือถือและคอมพิวเตอร์ จะได้ไม่ต้องมาผูกติดกับมือถือยี่ห้อใดยี่ห้อหนึ่งอีก

ลอง Google Keep ดูครับ เพราะอย่างน้อยก็ใช้ได้บนหลายอุปกรณ์

แต่โห... รูปแบบการจัดการบันทึกน้อยไปหน่อยสำหรับผมครับ จดได้ก็จริง แต่จัดเก็บและใช้งานไม่ได้อย่างที่ต้องการ โน้ตไหนจด bullet แล้วก็ต้องมีแค่ bullet บ้าไปแล้ว!

ลองหาแอปอื่นอยู่พักหนึ่ง สุดท้ายก็เกิดความคิดขึ้นมาว่า...

สร้างเองซะเลยดีไหม?



จริง ๆ ผมเขียนโปรแกรมเป็นอยู่แล้วครับ ถ้าจะลงมือเขียนเว็บแอปเองก็ทำได้ (ถึงแม้ที่ถนัดจริง ๆ จะเป็นสาย Automation ก็ตาม)

แต่เดี๋ยวก่อน...

นี่มันยุค AI แล้วนะ!

ในเมื่อเรามี AI ที่ช่วยเขียนโปรแกรมได้ ทำไมผมต้องกลับไปนั่งเขียนโค้ดเองอีกล่ะ?

ผมเลยตั้งโจทย์ให้ตัวเองขึ้นมาข้อหนึ่งครับ

โปรเจกต์นี้ผมจะไม่เขียนโค้ดเองเลย ขอเกาะเทรนด์ซะหน่อย~

ผมจะทำหน้าที่แค่บอกว่าอยากได้อะไร ให้ AI เป็นคนเขียนโปรแกรม แก้ไข และปรับแต่งทั้งหมด ส่วนผมจะคอยดูผลลัพธ์ ทดลองใช้งาน และบอกว่าตรงไหนยังไม่ถูกใจ

ขอขึ้นมาเป็น Project Manager กับเขาบ้างเถอะครับ!

คราวนี้จะขอสั่งงานอย่างเดียว ไม่ลงไปดูโค้ดเลย 555+

แต่อย่างที่บอกตอนต้น เหตุการณ์นี้มันเกิดเมื่อสัก 2 ปีที่แล้ว (2024)

ผลเป็นยังไงน่ะเหรอ?

พังพินาศ! 555+



ตอนนั้น AI Chatbot กำลังเป็นที่ฮือฮามาก ซึ่งก็ต้องยอมรับว่าเก่งจริงครับ เขียนโค้ดได้ ช่วยคิดได้ ตอบคำถามเกี่ยวกับโปรแกรมได้สารพัด

แต่การจะให้ AI สร้างเว็บแอปทั้งระบบ โดยที่เราไม่ต้องลงมือเขียนโค้ดเองเลย ยังไม่ใช่เรื่องง่ายในตอนนั้นครับ

AI ในตอนนั้นยังเขียนโปรแกรมได้ไม่สมบูรณ์พอสำหรับงานลักษณะนี้ และที่สำคัญคือยังไม่สามารถเข้ามาทำงานกับไฟล์โปรเจกต์บนเครื่องของเราได้โดยตรง

เวลาจะให้แก้อะไร ก็ต้องคัดลอกโค้ดไปวางในแชต รอให้ AI แก้ แล้วคัดลอกกลับมาใส่ในไฟล์เอง

แก้จุดหนึ่งก็ต้องคัดลอกไปมาอีก ทำไปทำมากลายเป็นว่าผมต้องกลับมายุ่งกับโค้ดอยู่ดี

ซึ่งผิดจากโจทย์ที่ตั้งไว้ตั้งแต่แรกเลยครับ

แต่ผมก็ไม่ได้เลิกคิดเรื่องนี้นะครับ

เวลาว่าง ๆ ผมก็กลับมาลองใหม่เป็นระยะ ดูว่า AI พัฒนาไปถึงไหนแล้ว มีเครื่องมืออะไรใหม่ ๆ ที่จะช่วยให้ทำแบบที่คิดไว้ได้บ้าง

จนเวลาผ่านไปปีสองปี...

แล้วสิ่งที่ผมรอก็มาถึงครับ

AI Agent ที่สามารถเข้ามาทำงานกับไฟล์โปรเจกต์บนเครื่องของเราได้จริง ๆ

คราวนี้ต่างจากเดิมมากครับ

AI สามารถอ่านไฟล์ในโปรเจกต์ สร้างไฟล์ใหม่ แก้โค้ดหลายส่วนที่เกี่ยวข้องกัน และช่วยตรวจสอบปัญหาได้ โดยที่เราไม่ต้องคอยคัดลอกโค้ดไปมาทีละก้อนเหมือนเมื่อก่อน

ความสามารถในการเขียนโปรแกรมของ AI เองก็พัฒนาไปไกลมาก จนสามารถสร้างเว็บแอปทั้งระบบ ปรับหน้าตา เพิ่มฟังก์ชัน และแก้ไขข้อผิดพลาดได้ดีขึ้นอย่างเห็นได้ชัด คือทำงานระดับ Senior Programmer เลยมั้ง!?

ผมก็เลยกลับมาลองทำตามโจทย์เดิมอีกครั้งครับ

ไม่เขียนโค้ดเอง ไม่เข้าไปนั่งไล่อ่านโค้ดเพื่อแก้ปัญหา ไม่แม้แต่จะเปิดไฟล์ดูด้วยซ้ำ! ขอสั่งและตรวจงานก็พอครับ ไม่ผ่านก็สั่งให้กลับไปแก้

และในที่สุด...

ผมก็ได้เว็บแอปจดโน้ตที่ต้องการมาใช้จริง ๆ ครับ!

ต่อไปผมอยากจดอะไรก็จด โยนเต็มที่ เปิดจากที่ไหนก็ได้ ไม่ต้องกลัวมือถือเสีย ... มีความสุข 😊 (1000+ โน้ตแล้วครับ ล่าสุด 555+)



สำหรับผม เรื่องที่น่าตื่นใจไม่ได้มีแค่การได้เว็บแอปจดโน้ตนะครับ

แต่เป็นการพิสูจน์ว่า วิธีทำงานที่ผมคิดไว้ตั้งแต่สองปีก่อน **สามารถทำได้จริงแล้ว**

ผมไม่ได้ต้องการพิสูจน์ว่า AI เขียนโค้ดเก่งกว่าโปรแกรมเมอร์ หรือคนเขียนโปรแกรมไม่จำเป็นอีกต่อไปนะครับ

สิ่งที่ผมอยากรู้มาตลอดคือ ถ้าเรารู้ว่าต้องการอะไร สามารถอธิบายงานให้ชัดเจน และตรวจสอบผลลัพธ์ได้ เราจะสามารถสร้างโปรแกรมที่ใช้งานได้จริง โดยไม่ต้องลงมือเขียนโค้ดเองเลยหรือเปล่า?

ตอนนี้ผมได้คำตอบแล้วครับ

และที่สำคัญ วิธีนี้ไม่ได้จำกัดอยู่แค่เว็บแอปจดโน้ตด้วย

เพราะหลังจากนั้น ผมก็มีเรื่องที่อยากทำอีกแล้วครับ...

ผมอยากได้เว็บแอปบริหารการเงินสำหรับร้านข้าวแกง! 555+

คราวนี้ไม่ใช่แค่ทำเว็บแอปไว้จดข้อมูลส่วนตัวแล้วนะครับ แต่เป็นเว็บแอปที่ต้องนำมาใช้จัดการการเงินจริง ๆ ของร้าน

คุณสมบัติที่ต้องการเป็นพิเศษคือ **ต้องสามารถบันทึกวันที่ได้ 2 แบบใน 1 รายการ**

เช่น **รายการซื้อหมูบด 10 กก.**

- Date: ประจำวันที่ 10 ส.ค. (วันที่หมูถูกทำขาย เช่น ทำเป็นต้มจืดหมูสับ)
- Cash_date: จ่ายเงินวันที่ 8 ส.ค. (ซื้อมาสต๊อก 2 วันก่อนทำอาหารจริง)

**Date และ Cash_date เป็นชื่อที่ผมตั้งขึ้นมาเองนะครับ ไม่มีวิชาการอะไรหรอก

ตอนนั้นผมก็คิดขึ้นมาว่า อืม... มีอะไรบางอย่าง คล้ายๆแอปจดโน้ตผมเลยแฮะ แอปจดโน้ตมีเป็นพัน แอปบัญชีมีเป็นร้อย แต่เขาไม่ได้สร้างมาเพื่อเรา บางแอปอาจจะมีฟังชั่นนี้นะ แต่ก็มีฟังชั่นอีก 108 ใช้ยากอะ แต่ละแอปก็มีข้อจำกัดแปลกๆที่ทำให้เราไม่ถูกใจ แสดงว่าเราไม่ได้ต้องการแอปแปลกใหม่อะไรสิ เราแค่ต้องการแอปที่มันตรงการใช้งานเราจริงๆ

ผมจึงเริ่มสร้างเว็บแอปตัวใหม่นี้ด้วยวิธีเดียวกัน และระหว่างที่ลงมือทำ ผมก็เริ่มสังเกตเห็นอะไรบางอย่างอีก

ถึงเว็บแอปจดโน้ตกับเว็บแอปบริหารการเงินจะทำงานคนละเรื่องกัน แต่ขั้นตอนหลายอย่างที่ใช้สร้างกลับเหมือนกันแฮะ

เราต้อง

- ออกแบบหน้าตา
- เตรียมระบบพื้นฐาน
- เชื่อมฐานข้อมูล
- ทำให้ฟังก์ชันต่าง ๆ ใช้งานได้
- แล้วจึงนำเว็บขึ้นออนไลน์

อ้าว มีลำดับงานชัดเจนเลยนี่นา

ผมเริ่มคิดว่า ถ้าอย่างนั้น ทำไมเราต้องกลับมาเริ่มคิดและทำขั้นตอนเหล่านี้ใหม่ทุกครั้งด้วยล่ะ?

น่าจะมีวิธีจัดกระบวนการทั้งหมดให้เป็นลำดับ และเก็บสิ่งที่ใช้ซ้ำได้ไว้เป็นแม่แบบสิ

จากนั้นผมก็เริ่มทดลองปรับวิธีทำงานของตัวเองครับ

สร้าง **เว็บแอปใหม่หนึ่งตัว** และสร้าง **วิธีสร้างเว็บแอปหนึ่งตัว** ไปพร้อม ๆ กันเลย กลับมาดูว่าขั้นตอนไหนใช้ซ้ำได้ ขั้นตอนไหนไม่จำเป็นต้องทำใหม่ และตรงไหนที่ควรเตรียมไว้ตั้งแต่ต้น เพื่อไม่ต้องมาเสียเวลาแก้ทีหลัง

พอทำซ้ำหลายครั้ง กระบวนการก็ค่อย ๆ ชัดเจนขึ้น และการสร้างเว็บแอปตัวต่อไปก็ง่ายขึ้นเรื่อย ๆ

จนสุดท้ายผมมีวิธีทำงานที่สามารถหยิบมาใช้กับเว็บแอปใหม่ ๆ ได้ โดยไม่ต้องกลับไปเริ่มคิดทุกอย่างตั้งแต่ศูนย์อีกแล้วครับ

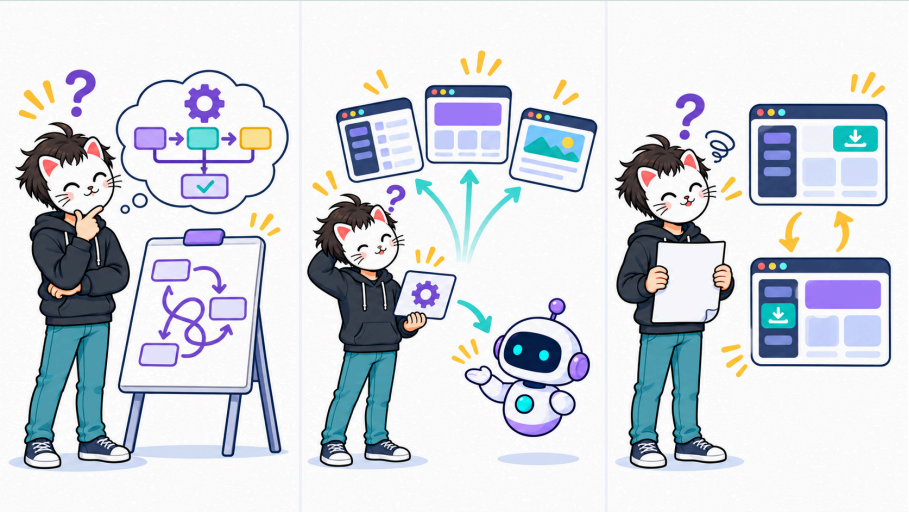
ตอนนี้มีเว็บแอปไว้ใช้งานเพียบเลยครับ 555+

และนี่แหละครับ คือจุดเริ่มต้นของหนังสือเล่มนี้

ผมอยากนำวิธีที่ทดลองใช้จริงจนได้ผล มาเรียบเรียงให้คนที่ไม่เคยเขียนโปรแกรมมาก่อนสามารถนำไปทำตามได้

แต่พอเริ่มลงมือเขียนหนังสือจริง ๆ ผมกลับพบว่า...

การสร้างเว็บแอปเอง กับการสอนให้คนอื่นสร้างตาม เป็นคนละเรื่องกันเลยครับ! 555+



ผมเจอปัญหาใหญ่อยู่ 3 เรื่อง

1. จะอธิบายสิ่งที่อยู่ในหัวออกมายังไงดี?

การสอนคนอื่นนี่ดูเหมือนจะง่ายนะครับ เพราะผมก็สร้างเว็บแอปมาเองกับมือ แถมทำมาหลายตัวแล้วด้วย

แต่พอจะเขียนอธิบายให้คนอื่นทำตามจริง ๆ กลับไม่ง่ายแฮะ

เวลาผมทำงานเอง ผมรู้อยู่แล้วว่าต้องทำอะไรก่อน อะไรหลัง ต้องเตรียมอะไรบ้าง และแต่ละขั้นตอนควรได้ผลลัพธ์แบบไหน

บางอย่างทำจนเคยชินไปแล้วด้วยซ้ำครับ

แต่สำหรับคนที่ไม่เคยสร้างเว็บแอปมาก่อน เขาไม่ได้มีภาพทั้งหมดอยู่ในหัวเหมือนผมนี่นา

อย่างผมอาจเขียนสั้น ๆ ว่า “เปิดโฟลเดอร์โปรเจกต์ในโปรแกรมที่เราใช้สร้างเว็บแอป” แล้วก็ไปขั้นตอนต่อไปเลย

แต่เดี๋ยวก่อน...

คนอ่านอาจยังไม่รู้ด้วยซ้ำว่าโฟลเดอร์โปรเจกต์คืออะไร ต้องสร้างไว้ตรงไหน แล้วเปิดในโปรแกรมยังไง!

อ้าว ถ้าอย่างนั้นผมก็ต้องย้อนกลับมาอธิบายเรื่องพวกนี้ด้วยสิ

พอเริ่มไล่ดูจริง ๆ ก็พบว่ามีรายละเอียดเล็ก ๆ แบบนี้เต็มไปหมดครับ

ผมเลยต้องกลับมาคิดใหม่ว่า จะเรียงลำดับการสอนยังไงดี อะไรที่ควรอธิบายตั้งแต่ต้น อะไรที่ยังไม่จำเป็นต้องรู้ และตรงไหนควรหยุดให้คนอ่านลองทำและตรวจผลก่อน

เพราะเป้าหมายของผมไม่ใช่แค่เขียนให้ครบทุกขั้นตอน แต่ต้องทำให้คนที่ไม่เคยเขียนโปรแกรมมาก่อน **อ่านแล้วทำตามได้จริง ๆ** ครับ

2. ใช้ AI เหมือนกัน แต่ทำไมผลลัพธ์ไม่เหมือนกัน?

เรื่องนี้ยิ่งน่าปวดหัวครับ 555+

ปกติหนังสือสอนใช้โปรแกรมทั่วไป เราอาจบอกว่าให้เปิดเมนูนี้ กดปุ่มนั้น เลือกคำสั่งนี้ แล้วทุกคนก็จะได้ผลลัพธ์ใกล้เคียงกัน

แต่หนังสือเล่มนี้เราใช้ AI เป็นคนสร้างเว็บแอปให้ครับ

สมมติผมให้ Prompt ตัวอย่างไปหนึ่งชุด คนอ่านสิบคนเอาไปใช้เหมือนกันทุกตัวอักษร ก็ไม่ได้หมายความว่าจะได้เว็บแอปหน้าตาเหมือนกันทั้งสิบคน

บางคนอาจได้ปุ่มอยู่ด้านบน บางคนได้ปุ่มอยู่ด้านล่าง บางคนได้ไฟล์แยกกันหลายไฟล์ ขณะที่อีกคน AI อาจจัดโครงสร้างไฟล์ออกมาอีกแบบหนึ่ง

ทั้งที่ใช้คำสั่งเดียวกันแท้ ๆ!

แล้วถ้าคนอ่านไม่ได้อยากสร้างเว็บแอปการเงินเหมือนตัวอย่างของผม แต่อยากสร้างเว็บแอปบันทึกงาน จัดการสินค้า หรือทำอะไรอย่างอื่นอีกล่ะ?

คราวนี้ผลลัพธ์ก็ยิ่งต่างกันไปใหญ่ครับ

ผมเลยมีคำถามว่า...

จะสอนยังไงให้คนอ่านทำต่อได้ แม้สิ่งที่ AI สร้างออกมาจะไม่เหมือนตัวอย่างในหนังสือ?

เพราะผมไม่อยากให้หนังสือเล่มนี้ใช้ได้เฉพาะกับเว็บแอปตัวอย่าง หรือใช้ได้เฉพาะตอนที่ AI ตอบออกมาตรงกับภาพหน้าจอที่ผมเตรียมไว้เท่านั้นครับ

3. AI กับเครื่องมือเปลี่ยนเร็วเหลือเกิน!

เรื่องสุดท้ายนี่เป็นปัญหาที่ผมควบคุมไม่ได้เลยครับ

AI และเครื่องมือที่เราใช้พัฒนาเร็วมาก

วันนี้ผมเปิดโปรแกรมขึ้นมา เมนูอยู่ตรงนี้ ปุ่มอยู่ตรงนั้น ผมก็ถ่ายภาพหน้าจอมาใส่หนังสือ พร้อมเขียนอธิบายไว้อย่างดี

อีกไม่กี่เดือน โปรแกรมอัปเดตใหม่...

อ้าว ปุ่มหายไปไหนแล้ว!?

บางทีไม่ได้หายหรอกครับ แค่ย้ายไปอยู่ที่อื่น หรือเปลี่ยนชื่อเมนูใหม่

บางบริการก็เปลี่ยนวิธีสมัครสมาชิก วิธีตั้งค่า หรือขั้นตอนการเชื่อมต่อไปเลย

ส่วน AI เองก็มีโมเดลใหม่ ๆ ออกมาเรื่อย ๆ ความสามารถและวิธีทำงานก็เปลี่ยนไปด้วย

ผมคงกลับมาแก้หนังสือทุกครั้งที่มีใครย้ายปุ่มไม่ได้หรอกครับ 555+

แล้วลองคิดในมุมคนอ่านดูนะครับ

ถ้าเปิดหนังสือมาแล้วเห็นว่าให้กดปุ่มหนึ่ง แต่หน้าจอของตัวเองไม่มีปุ่มนั้น จะทำยังไงต่อ?

ทั้งที่จริง ๆ เป้าหมายของขั้นตอนนั้นอาจยังเหมือนเดิมทุกอย่าง เพียงแต่วิธีทำเปลี่ยนไปแล้วเท่านั้นเอง

นี่จึงเป็นอีกเรื่องที่ผมต้องหาทางรับมือครับ

พอมองปัญหาทั้งสามข้อรวมกัน ผมก็เริ่มเห็นว่า ถ้าจะสอนด้วยวิธีบอกให้กดตรงนั้น กดตรงนี้ แล้วหวังว่าทุกคนจะได้ผลลัพธ์เหมือนกัน หนังสือเล่มนี้คงไปได้ไม่ไกลแน่ครับ

เพราะทั้งพื้นฐานของคนอ่าน ผลลัพธ์จาก AI และหน้าตาของเครื่องมือ ล้วนมีโอกาสแตกต่างจากตัวอย่างที่ผมเตรียมไว้

ผมจึงต้องหาวิธีสอนที่ทำให้คนอ่านเข้าใจงานที่กำลังทำ ไม่ใช่แค่จำขั้นตอนตามภาพหน้าจอ

เมื่อคิดได้แบบนี้...

ผมจึงเลือกใช้หลักการ Concept Art เป็นหัวใจของหนังสือเล่มนี้ครับ

ในแต่ละบท ผมจะเริ่มด้วยภาพสรุปลำดับงานทั้งหมดให้ดูก่อน ว่าเรากำลังจะทำอะไร ต้องผ่านขั้นตอนไหนบ้าง และสุดท้ายควรได้ผลลัพธ์แบบไหน

ภาพนี้ใช้เป็น Guide ให้เราเห็นภาพรวมของงานได้เลยครับ และแน่นอนว่าผมมี Concept Art ที่สรุปกระบวนการทั้งหมดในหนังสือเล่มนี้ไว้ในภาพเดียวด้วยเช่นกัน

เรียกว่าภาพเดียวแทนหนังสือทั้งเล่มเลยครับ 555+

เมื่อเห็นภาพรวมแล้ว เราก็ค่อยมาลงมือทำจริงทีละขั้นตอน โดยมีภาพหน้าจอและคำอธิบายประกอบให้ทำตาม

แต่มีเรื่องหนึ่งที่ผมอยากให้คุณให้ความสำคัญครับ

สิ่งสำคัญไม่ใช่การทำให้หน้าจอออกมาเหมือนในหนังสือทุกจุด แต่เป็นการเข้าใจว่าเรากำลังทำอะไร และตรวจสอบได้ว่างานนั้นสำเร็จตามเป้าหมายหรือยัง

ยกตัวอย่างเช่น ในขั้นตอนการเตรียม AI ให้ทำงานกับไฟล์โปรเจกต์ของเรา ผมอาจสรุปลำดับงานไว้ว่า

สมัครสมาชิก → นำบัญชีมาเชื่อมต่อ → ตรวจสอบว่า AI สามารถทำงานกับไฟล์ของเราได้

ทีนี้สมมติว่าคุณกำลังทำตามหนังสืออยู่ แต่หน้าจอโปรแกรมไม่เหมือนในตัวอย่าง ปุ่มเชื่อมต่อย้ายไปอยู่ที่อื่น หรือมีขั้นตอนยืนยันตัวตนเพิ่มเข้ามา

แล้วจะทำยังไงดี?

ก็กลับมาดู Concept Art ของเราครับ ว่าตอนนี้กำลังทำขั้นตอนไหนอยู่

อ๋อ... ติดตรงสมัครสมาชิกใช่ไหม?

ถ้าหน้าจอไม่เหมือนที่ผมสอน ก็ลองทำตามหน้าจอที่เห็นได้เลยครับ เรื่องนี้ไม่เกินความสามารถของทุกคนหรอก

หรือจะเปิด Google AI แล้วถามว่า “ช่วยบอกวิธีสมัครสมาชิกเว็บไซต์นี้อย่างละเอียดหน่อย” ก็ได้ครับ

พอสมัครสำเร็จ เราก็กลับมาดูขั้นตอนถัดไปใน Concept Art แล้วทำต่อได้เลย

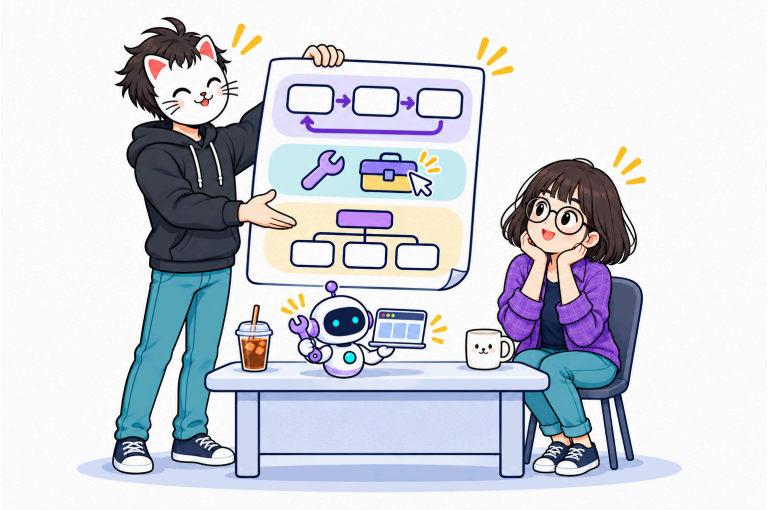
ไม่จำเป็นต้องทำเหมือนตัวอย่างทุกจุด ขอแค่ทำงานในขั้นตอนนั้นให้สำเร็จก็พอครับ

เพราะฉะนั้น ต่อให้ AI สร้างอะไรออกมาไม่เหมือนตัวอย่าง หรือโปรแกรมเปลี่ยนหน้าตาไป เราก็ยังรู้ว่าต้องการผลลัพธ์อะไร และสามารถให้ AI ช่วยหาวิธีทำต่อได้ครับ

สิ่งที่ผมอยากส่งต่อให้ผู้อ่าน eBook เล่มนี้

หลังจากทดลองสร้างเว็บแอปด้วย AI มาหลายตัว ผมก็ได้กระบวนการทำงานที่ใช้เองอยู่ทุกวันนี้ และอยากนำมาแบ่งปันผ่านหนังสือเล่มนี้ครับ

ผมตั้งใจไว้ 3 เรื่องหลัก ๆ



1. ลำดับงานจริงที่ทำซ้ำได้ ตั้งแต่กระดาษเปล่าจนเว็บแอปออนไลน์

ผมอยากถ่ายทอดวิธีสร้างเว็บแอปด้วย AI ที่ผมทดลองและใช้งานจริง ไม่ใช่แค่แนวคิดหรือวิธีที่อ่านมาจากที่อื่นครับ

เราจะเริ่มตั้งแต่ยังไม่มีอะไรเลย แล้วค่อย ๆ ทำไปตามลำดับ

- ออกแบบหน้าตาเว็บแอป
- เตรียมโครงสร้างพื้นฐาน
- เชื่อมต่อฐานข้อมูลออนไลน์
- ทำให้ฟังก์ชันต่าง ๆ ใช้งานได้จริง
- ทดลองและตรวจสอบผลลัพธ์
- นำเว็บแอปขึ้นออนไลน์

ทั้งหมดนี้เป็นลำดับงานเดียวกับที่ผมใช้สร้างเว็บแอปของตัวเอง และนำกลับมาใช้ซ้ำกับโปรเจกต์ใหม่ ๆ ได้ครับ

2. เครื่องมือและสภาพแวดล้อมที่ผมคิดว่าเหมาะกับงานนี้จริง ๆ

ทุกวันนี้มีเครื่องมือสำหรับสร้างเว็บแอปด้วย AI ให้เลือกเยอะมากครับ เยอะจนบางทีไม่รู้จะเลือกอะไรดี 555+

บางตัวสร้างเว็บได้เร็ว บางตัวใช้งานง่าย บางตัวมีฟังก์ชันสารพัดอย่าง

แต่ผมไม่ได้เลือกจากความนิยมหรือเลือกเพราะดูง่ายที่สุดอย่างเดียวครับ

ผมเลือกจากคำถามว่า **เครื่องมือชุดไหนจะช่วยให้เราสร้างเว็บแอปตั้งแต่ต้นจนจบและนำไปต่อยอดได้จริง?**

ผมจึงเลือกชุดเครื่องมือและจัดสภาพแวดล้อมการทำงานที่คิดว่าเหมาะกับกระบวนการนี้ โดยดูทั้งความสามารถ ความยืดหยุ่น และการนำไปใช้งานจริงในระยะยาว

แน่นอนครับ นี่เป็นการเลือกจากประสบการณ์และมุมมองของผมเอง ไม่ได้หมายความว่าเครื่องมือชุดนี้ดีที่สุดสำหรับทุกคน

แต่เป็นสภาพแวดล้อมที่ผมใช้ทำงานจริง และคิดว่าเหมาะสำหรับถ่ายทอดวิธีสร้างเว็บแอปในหนังสือเล่มนี้ครับ

ส่วนจะใช้เครื่องมืออะไรบ้าง และต้องจัดสภาพแวดล้อมการทำงานยังไง ติดตามได้ในเล่มครับ (ยิ้มอ่อน)

3. Concept Art ที่ช่วยให้เห็นภาพรวมและไม่หลงทาง

เรื่องนี้เป็นสิ่งที่ผมให้ความสำคัญมากตอนเขียนหนังสือครับ

เพราะผมรู้ดีว่า คนที่ไม่เคยสร้างเว็บแอปมาก่อน อาจยังนึกภาพไม่ออกด้วยซ้ำว่าแต่ละขั้นตอนเกี่ยวข้องกันยังไง และทำไปเพื่ออะไร

ผมจึงพยายามนำงานแต่ละช่วงมาจัดเป็น Concept Art ให้เห็นภาพรวมของสิ่งที่กำลังจะทำ ตั้งแต่ลำดับขั้นตอน ไปจนถึงผลลัพธ์ที่ควรได้

จากนั้นจึงค่อยพาลงมือทำจริง พร้อมภาพหน้าจอ ตัวอย่าง และคำอธิบายประกอบ

ผมอยากให้ภาพเหล่านี้เป็นเหมือนแผนที่ที่คุณสามารถเปิดกลับมาดูได้เสมอครับ

ต่อให้วันหนึ่งหน้าตาโปรแกรมเปลี่ยนไป หรือรายละเอียดทางเทคนิคไม่เหมือนในหนังสือ อย่างน้อยคุณก็ยังรู้ว่ากำลังทำอะไร อยู่ตรงไหนของงาน และต้องไปต่อทางไหน

เพราะผมไม่ได้อยากให้คุณแค่ทำตามหนังสือได้ แต่อยากให้เข้าใจกระบวนการจนสามารถนำไปใช้กับเว็บแอปตัวต่อ ๆ ไปได้ด้วยครับ

นี่คือ 3 เรื่องที่ผมตั้งใจถ่ายทอดผ่านหนังสือเล่มนี้ครับ

ลำดับงานจริงที่ทำซ้ำได้ เครื่องมือที่เลือกมาให้เหมาะกับงาน และ Concept Art ที่ช่วยให้คุณเห็นภาพรวมและเดินตามลำดับงานได้อย่างมั่นใจ

ทั้งหมดนี้มาจากประสบการณ์ที่ผมลงมือทำ ทดลอง และปรับปรุงด้วยตัวเองครับ

ผมหวังว่าประสบการณ์เหล่านี้จะช่วยให้คุณสร้างเว็บแอปที่อยากได้ขึ้นมาสักตัวหนึ่ง และเมื่อทำสำเร็จแล้ว ก็สามารถนำวิธีเดียวกันไปสร้างตัวต่อไปได้อีก

เหมือนที่ผมทำอยู่ทุกวันนี้ครับ

ขอให้สนุกกับการสร้างเว็บแอปของตัวเองนะครับ!