



ใช้ ChatGPT

THE
BOOK
BRAINS

มือใหม่
ก็ทำได้

ไม่ต้องเรียน
โปรแกรม

สร้างแอป มือถือน ด้วย AI

สำหรับคนไม่เขียนโค้ด

ใช้ ChatGPT สร้างแอปได้
แม้เริ่มจากศูนย์



สั่ง AI ด้วยคำสั่ง (Prompt)
สร้างแอปตามที่ต้องการ



เชื่อมต่อฐานข้อมูลได้จริง
ปลอดภัย ใช้งานได้จริง



ทำแอปได้ทั้ง
Android และ iPhone



ทำตามได้ง่าย
ทีละขั้นตอน



มีตัวอย่างจริง
พร้อมใช้งาน



พร้อม Prompt
ใช้งานได้ทันที



NOVA
SCIENCE

ผู้จัดทำ : ศักดิ์สุตา ฉันทเตยานนท์

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537

ห้ามทำซ้ำ คัดลอก ดัดแปลง เผยแพร่โดยไม่ได้รับอนุญาต

สารบัญ

บทนำ	จากไอเดียสู่แอปด้วย ChatGPT	7
-------------	-----------------------------------	---

PART 1 — เริ่มต้น

บทที่ 1	ยุคที่การสร้างแอปไม่ได้เริ่มจากการเขียนโค้ด	13
บทที่ 2	รู้จักเครื่องมือที่ต้องใช้	19
บทที่ 3	เปลี่ยนไอเดียให้เป็นแผนแอป	29
บทที่ 4	คุยกับ ChatGPT อย่างไรให้ได้ Code ที่ใช้ได้	42

PART 2 — สร้างแอปแรก

บทที่ 5	สร้าง Flutter Project แรก	57
บทที่ 6	สร้างหน้าจอและ UI	73
บทที่ 7	ทำให้ปุ่มและฟังก์ชันทำงาน	96

บทที่ 8 เก็บข้อมูลด้วย Database	115
บทที่ 9 Debug เมื่อแอปเกิดปัญหา	153

PART 3 — สร้างแอปจริง

บทที่ 10 โปรเจกต์: แอปรายรับรายจ่าย MoneyNote	172
บทที่ 11 โปรเจกต์: To-do List	227
บทที่ 12 โปรเจกต์: แอปร้านค้า	257
บทที่ 13 โปรเจกต์: AI Chatbot	300

PART 4 — จาก Prototype สู่แอปจริง

บทที่ 14 Login, API และระบบที่ต้องรู้	343
บทที่ 15 Security และการทดสอบ	406
บทที่ 16 Build แอป Android และ iPhone	450
บทที่ 17 นำแอปขึ้น Store	486
บทที่ 18 เปลี่ยนแอปให้กลายเป็นรายได้	530

บทส่งท้าย จากคนไม่เขียนโค้ด สู่คนที่สร้างแอปด้วย AI 588

บทนำ

จากไอเดียสู่แอปด้วย ChatGPT

ถ้าคุณมีไอเดียอยากสร้างแอป แต่ไม่เคยเขียนโค้ดมาก่อน คุณอาจคิดว่าเรื่องนี้ไกลตัวเกินไป

ต้องเรียน Programming

ต้องรู้เรื่อง Database

ต้องเข้าใจ API

ต้องมี Developer

และอาจต้องใช้เวลาหลายเดือนกว่าจะสร้างแอปได้จริง

แต่วันนี้วิธีการสร้างแอปกำลังเปลี่ยนไป

AI อย่าง ChatGPT สามารถช่วยให้คุณวางแผน ออกแบบ เขียนโค้ด อธิบายโค้ด และช่วยหาสาเหตุเมื่อแอปเกิดปัญหาได้

นั่นไม่ได้หมายความว่า你不ต้องเรียนรู้อะไรเลย

แต่หมายความว่า คุณไม่จำเป็นต้องเริ่มต้นด้วยการเป็นโปรแกรมเมอร์

สิ่งที่คุณต้องมีคือไอเดีย ปัญหาที่ต้องการแก้ และความสามารถในการอธิบายสิ่งที่ต้องการให้ AI เข้าใจ

หนังสือเล่มนี้จะพาคุณทำอะไร

เราจะเริ่มจากศูนย์

ตั้งแต่การเปลี่ยนไอเดียให้เป็นแผนของแอป การเขียน Prompt การสร้างหน้าจอ การเขียนโค้ดด้วย AI การจัดเก็บข้อมูล การแก้ Error ไปจนถึงการเตรียมแอปสำหรับใช้งานจริง

ระหว่างทาง คุณจะได้เห็นตัวอย่างจากแอปหลายรูปแบบ เช่น

- แอปรายรับรายจ่าย
- To-do List
- แอปร้านค้า
- AI Chatbot

แต่เป้าหมายไม่ใช่การทำให้คุณจำโค้ดเหล่านี้

เป้าหมายคือทำให้คุณเข้าใจ กระบวนการสร้างแอปด้วย AI

เมื่อเข้าใจกระบวนการแล้ว คุณสามารถนำวิธีเดียวกันไปสร้างแอปประเภทอื่นได้

คุณไม่จำเป็นต้องรู้โค้ดทุกบรรทัด

นี่เป็นเรื่องที่อยากให้เข้าใจก่อนเริ่มอ่าน

หนังสือเล่มนี้ไม่ได้บอกว่า AI จะสร้างแอปให้คุณด้วย Prompt เดียว

ในโลกจริง AI สามารถเขียนโค้ดผิดได้ เข้าใจ Requirement ผิด หรือสร้างโค้ดที่ใช้ไม่ได้กับโปรเจกต์ของคุณ

ดังนั้นทักษะที่สำคัญไม่ใช่แค่

“สั่ง AI ให้เขียนโค้ด”

แต่คือ

“รู้ว่าจะให้ AI ทำอะไร ตรวจสอบอย่างไร และแก้ปัญหายังไงเมื่อสิ่งที่สร้างไม่ทำงาน”

คุณไม่จำเป็นต้องรู้ทุกอย่างตั้งแต่วันแรก

แต่เมื่ออ่านจบ คุณควรสามารถมองโค้ดแล้วพอเข้าใจว่ามันกำลังทำอะไร และสามารถใช่ AI ช่วยคุณเดินหน้าต่อได้

วิธีใช้หนังสือเล่มนี้ให้ได้ผลที่สุด

อย่าอ่านอย่างเดียว

เปิดคอมพิวเตอร์ไปพร้อมกัน

เมื่อเจอ Prompt ให้ลองใช้

เมื่อเจอโค้ดให้ลองรัน

เมื่อเกิด Error อย่าเพิ่งตกใจ ให้ลองใช้ ChatGPT ช่วยวิเคราะห์

เพราะการสร้างแอปเป็นทักษะที่เรียนรู้ได้ดีที่สุดจากการลงมือทำ

คุณไม่จำเป็นต้องเข้าใจทุกอย่างก่อนเริ่ม

เริ่มจากสิ่งเล็ก ๆ → ทดลอง → เจอปัญหา → แก้ → แล้วค่อยเพิ่ม
ความสามารถ

นี่คือวิธีที่เราจะใช้ตลอดทั้งเล่ม

สิ่งที่คุณจะได้เมื่ออ่านจบ

เมื่อเดินทางมาถึงหน้าสุดท้าย คุณอาจยังไม่ใช้โปรแกรมเมอร์มืออาชีพ

และนั่นไม่ใช่เป้าหมาย

แต่คุณควรสามารถ

- เปลี่ยนไอเดียให้เป็นแผนแอป
- ใช้ ChatGPT ช่วยสร้างโค้ด

- สร้างและปรับ UI
- จัดการข้อมูล
- Debug เมื่อเกิดปัญหา
- เชื่อมต่อ API
- ทดสอบแอป
- เตรียมแอปสำหรับ Android และ iPhone
- เข้าใจขั้นตอนการนำแอปไปเผยแพร่
- และมองเห็นแนวทางต่อยอดแอปให้กลายเป็นธุรกิจ

จากนั้นสิ่งที่เหลือคือการฝึกฝน

เพราะความสามารถในการสร้างแอปไม่ได้เกิดจากการจำโค้ดจำนวนมาก

แต่มาจากการรู้ว่า ต้องการสร้างอะไร และใช้เครื่องมือที่มีอยู่เพื่อทำให้มันเกิดขึ้นได้อย่างไร

เริ่มจากศูนย์ได้จริง

คุณอาจกำลังอ่านหนังสือเล่มนี้โดยไม่เคยเขียนโค้ดแม้แต่บรรทัดเดียว
ไม่เป็นไร

เราจะไม่เริ่มด้วยคำศัพท์ยาก ๆ หรือทฤษฎีจำนวนมาก

เราจะเริ่มจากคำถามง่ายที่สุด

“คุณอยากสร้างแอปอะไร?”

จากคำถามนั้น เราจะค่อย ๆ เปลี่ยนไอเดียให้กลายเป็นหน้าจอ

เปลี่ยนหน้าจอให้กลายเป็นโค้ด

เปลี่ยนโค้ดให้กลายเป็นแอป

และเปลี่ยนแอปต้นแบบให้กลายเป็นผลงานที่สามารถนำไปใช้งานจริงได้

นี่ไม่ใช่หนังสือที่จะสอนให้คุณเป็นโปรแกรมเมอร์

แต่มันจะสอนให้คุณ **ใช้ AI เป็นเครื่องมือในการสร้างสิ่งที่ก่อนหน้านี้
คุณอาจคิดว่าตัวเองสร้างไม่ได้**

ถ้าคุณพร้อมแล้ว เรามาเริ่มจากการเปลี่ยนวิธีคิดเรื่องการสร้างแอปกัน
ก่อน

เพราะในยุคของ AI คำถามอาจไม่ใช่

“ฉันเขียนโค้ดเป็นหรือยัง?”

แต่อาจเป็น

“ฉันมีไอเดียอะไรที่อยากสร้างขึ้นมาบ้าง?”

บทที่ 1

ยุคที่การสร้างแอปไม่ได้เริ่มจากการเขียนโค้ด

เมื่อก่อน ถ้าคุณอยากสร้างแอป คำถามแรกที่มีมักเกิดขึ้นคือ

“ฉันเขียนโค้ดเป็นหรือยัง?”

เพราะการสร้างแอปเกี่ยวข้องกับหลายเรื่อง ทั้งภาษาโปรแกรม Framework Database API และเครื่องมือสำหรับทดสอบและเผยแพร่

สำหรับคนทั่วไป สิ่งเหล่านี้อาจกลายเป็นกำแพงที่ทำให้ไอเดียจำนวน มากหยุดอยู่แค่ในหัว

แต่ AI กำลังเปลี่ยนจุดเริ่มต้นของการสร้างแอป

วันนี้ คุณไม่จำเป็นต้องเริ่มจากคำถามว่า

“ฉันเขียนโค้ดเป็นหรือยัง?”

แต่สามารถเริ่มจากคำถามว่า

“ฉันอยากให้แอปทำอะไร?”

จากนั้นใช้ AI ช่วยเปลี่ยนไอเดียนั้นให้กลายเป็นแผน โครงสร้าง และโค้ด

จากไอเดียสู่แผน

ลองสมมติว่าคุณอยากสร้างแอปบันทึกรายรับรายจ่าย

คุณอาจบอก AI ว่า

“ฉันอยากสร้างแอปมือถือสำหรับบันทึกรายรับรายจ่าย ผู้ใช้สามารถเพิ่มรายการ ระบุจำนวนเงิน เลือกประเภท และดูยอดรวมได้ ช่วยวางโครงสร้างแอปสำหรับมือใหม่ให้หน่อย”

จากคำอธิบายนี้ AI สามารถช่วยคุณคิดต่อได้ว่า

- แอปควรมีหน้าจออะไร
- ต้องมีฟังก์ชันอะไร
- ต้องเก็บข้อมูลอะไร
- ควรจัดโครงสร้างโปรเจกต์อย่างไร
- ควรเริ่มพัฒนาจากส่วนไหนก่อน

จากนั้นจึงค่อยสร้างทีละส่วน

นี่คือการเปลี่ยนแปลงสำคัญ

คุณไม่จำเป็นต้องเริ่มจากโค้ด

แต่เริ่มจาก ความต้องการที่ชัดเจน

AI ไม่ได้ทำทุกอย่างแทนคุณ

การใช้ AI สร้างแอปไม่ได้หมายความว่า

“พิมพ์ Prompt ครึ่งเดียว แล้วได้แอปพร้อมขาย”

AI อาจสร้างโค้ดผิด ใช้ Library ที่ไม่ตรงกับโปรเจกต์ หรือเข้าใจสิ่งที่คุณต้องการไม่ครบ

ดังนั้น คุณยังมีหน้าที่สำคัญ 4 อย่าง

1. บอก AI ให้ชัดว่าต้องการอะไร

ยิ่งความต้องการชัด AI ยิ่งช่วยได้ตรงขึ้น

2. ตรวจสอบสิ่งที่ AI สร้าง

อย่าคิดว่า Code ที่ AI สร้างถูกต้องเสมอ

3. ทดลองใช้งานจริง

สิ่งที่ดูถูกต้องใน Code อาจไม่ได้ทำงานตามที่ต้องการเมื่อเปิดแอปจริง

4. แก้ปัญหาเมื่อเกิด Error

เมื่อเกิดปัญหา คุณต้องรวบรวมข้อมูลและใช้ AI ช่วยวิเคราะห์สาเหตุ

คุณไม่จำเป็นต้องเขียนโค้ดทุกบรรทัดเอง

แต่ควรเข้าใจมากพอที่จะรู้ว่า

คุณกำลังให้ AI สร้างอะไร

และผลลัพธ์ที่ถูกต้องควรเป็นอย่างไร

5 ทักษะของคนสร้างแอปด้วย AI

เมื่อ AI ช่วยเขียนโค้ดได้ บทบาทของคนสร้างแอปไม่ได้หายไป

แต่เปลี่ยนไป

ทักษะ	สิ่งที่ต้องทำ
Product Thinking	รู้ว่าแอปกำลังแก้ปัญหอะไร และใครจะใช้
Prompting	อธิบายสิ่งที่ต้องการให้ AI เข้าใจ
Problem Solving	แบ่งปัญหาใหญ่เป็นงานเล็ก
Debugging	หาสาเหตุเมื่อสิ่งที่สร้างไม่ทำงาน
Testing	ตรวจว่าผลลัพธ์ตรงกับสิ่งที่ต้องการจริงหรือไม่

พุดง่าย ๆ คือ

AI ช่วยทำงาน แต่คุณต้องเป็นคนคิด ตัดสินใจ และตรวจสอบ

ทักษะเหล่านี้สำคัญกว่าการพยายามจำโค้ดทุกบรรทัด

เริ่มจากเวอร์ชันแรก ไม่ใช่แอปที่สมบูรณ์แบบ

มือใหม่จำนวนมากเริ่มต้นด้วยไอเดียที่ใหญ่เกินไป

อยากมีทุก Feature ตั้งแต่วันแรก

แต่ยิ่งแอปมีขนาดใหญ่ จำนวนสิ่งที่ต้องตัดสินใจและจำนวนปัญหาที่ต้องแก้ก็ยิ่งเพิ่มขึ้น

ในช่วงเริ่มต้น เป้าหมายของคุณไม่ควรเป็น

“สร้างแอปที่สมบูรณ์แบบ”

แต่ควรเป็น

“สร้างเวอร์ชันแรกที่ทำงานได้จริง”

ในบทที่ 3 เราจะเรียนรู้วิธีเลือกสิ่งที่จำเป็น และกำหนดขอบเขตของแอปให้ชัดเจนก่อนเริ่มสร้าง

สิ่งที่ต้องเปลี่ยนคือวิธีคิด

สิ่งที่ AI เปลี่ยนไม่ได้มีแค่ความเร็วในการเขียนโค้ด

แต่คือการลดกำแพงระหว่าง

ไอเดีย → Prototype

คุณอาจยังสร้างแอปที่สมบูรณ์แบบไม่ได้ในครั้งแรก

แต่วันนี้คุณสามารถสร้างเวอร์ชันแรกขึ้นมา ทดลองใช้ และเรียนรู้จากของจริงได้เร็วขึ้น

ดังนั้น อย่าคิดว่า

“ฉันไม่รู้โค้ด ดังนั้นฉันสร้างแอปไม่ได้”

ให้เปลี่ยนเป็น

“ฉันยังไม่รู้โค้ด แต่ฉันสามารถใช้ AI ช่วย และเรียนรู้เฉพาะสิ่งที่จำเป็นในระหว่างการสร้างได้”

คุณไม่จำเป็นต้องรู้ทุกอย่างก่อนเริ่ม

สิ่งสำคัญคือรู้ว่า

คุณกำลังพยายามสร้างอะไร
และขั้นตอนต่อไปควรทำอะไร

บทที่ 2

รู้จักเครื่องมือที่ต้องใช้

ก่อนเริ่มสร้างแอป เรามาทำความรู้จักเครื่องมือที่หนังสือเล่มนี้จะใช้กันก่อน

ไม่ต้องจำรายละเอียดทางเทคนิคทั้งหมดในตอนนี้ สิ่งสำคัญคือ **รู้ว่าแต่ละเครื่องมือมีหน้าที่อะไร**

เส้นทางหลักของเราคือ

ChatGPT → VS Code → Flutter → Firebase

และเมื่อถึงขั้น Build แอปสำหรับ Android หรือ iPhone จะมีเครื่องมือเพิ่มเติมเข้ามาเกี่ยวข้อง

2.1 ChatGPT — ผู้ช่วยในการสร้างแอป

ChatGPT คือเครื่องมือที่เราจะใช้ตลอดทั้งเล่ม

มันสามารถช่วยเราได้หลายด้าน เช่น

- วางแผนแอป
- ออกแบบ Feature
- สร้าง Code
- อธิบาย Code
- วิเคราะห์ Error
- ช่วยเขียน Test
- ตรวจสอบแนวทางการพัฒนา

แต่การทำงานจริงจะไม่ได้จบด้วย Prompt เพียงครั้งเดียว

เราจะทำงานเป็นขั้นตอน เช่น

กำหนดสิ่งที่ต้องการ → ให้ AI ช่วยสร้าง → นำไปทดลอง → ตรวจสอบผล
→ แก้ปัญหา → ทดลองใหม่

ดังนั้น ChatGPT ทำหน้าที่เป็น **ผู้ช่วยในการคิดและพัฒนา** ส่วนเรายังเป็นคนกำหนดว่าต้องการสร้างอะไรและตรวจสอบผลลัพธ์ที่ได้

2.2 VS Code — โปรแกรมสำหรับการจัดการ Project

VS Code (Visual Studio Code) คือโปรแกรมที่เราใช้เปิดและจัดการ Project รวมถึงไฟล์ Code

ให้เข้าใจง่าย ๆ ว่า

ChatGPT ช่วยสร้างหรือแนะนำ Code

VS Code คือที่ที่เราเปิด Project และนำ Code ไปใช้งาน

ตัวอย่างเช่น ถ้า ChatGPT บอกว่า

ให้แก้ไขไฟล์ main.dart

เราจะเปิด Project ใน VS Code แล้วหาไฟล์ main.dart จากนั้นจึงแก้ไขตามคำแนะนำ

สิ่งที่คุณต้องเรียนรู้จึงไม่ใช่การจำ Code ทุกบรรทัด แต่คือการรู้ว่า ไฟล์ไหนทำหน้าที่อะไร และต้องแก้ตรงไหน

2.3 Flutter — Framework สำหรับสร้าง

แอป

Flutter คือ Framework สำหรับสร้างแอป

Framework หมายถึงชุดโครงสร้างและเครื่องมือที่ช่วยให้นักพัฒนาสร้างซอฟต์แวร์ได้ง่ายขึ้น โดยไม่ต้องสร้างทุกอย่างตั้งแต่ศูนย์

ในหนังสือเล่มนี้ เราจะใช้ Flutter เป็น Framework หลักสำหรับสร้าง Mobile App

ภาพรวมคือ

Flutter

↓

Mobile App

└─ Android

└─ iPhone

ข้อดีคือเราสามารถใช้นโยบายเดียวกันในการสร้างแอปสำหรับหลายแพลตฟอร์ม แทนที่จะต้องเริ่มเรียนเทคโนโลยีคนละชุดตั้งแต่ต้น

2.4 Firebase — ระบบเบื้องหลังของแอป

แอปบางประเภทสามารถทำงานโดยไม่ต้องมีระบบออนไลน์หรือฐานข้อมูลภายนอก

แต่ถ้าแอปต้องการสิ่งต่าง ๆ เช่น

- Login
- บัญชีผู้ใช้
- Database
- การเก็บข้อมูลบน Cloud

เราจะต้องมีระบบที่ทำงานอยู่เบื้องหลัง

หนังสือเล่มนี้เลือกใช้ **Firestore** สำหรับงานเหล่านี้

ส่วนสำคัญที่เราจะใช้คือ

Authentication

ใช้สำหรับระบบ Login และการจัดการผู้ใช้

Database

ใช้สำหรับจัดเก็บข้อมูลของแอป

ตัวอย่างเช่น แอปรับรายจ่ายอาจมีโครงสร้างแบบนี้

ผู้ใช้

↓

รายการรายรับรายจ่าย

↓

Firestore Database

ดังนั้น Firestore จะทำหน้าที่เป็นส่วนหนึ่งของ **Backend** หรือระบบเบื้องหลังของแอป

2.5 Android Studio และ Android SDK

แม้เราจะใช้ VS Code เป็นหลัก แต่เมื่อพัฒนาและ Build แอป Android เราจะเกี่ยวข้องกับ **Android Studio** และ **Android SDK**

Android Studio คืออะไร?

Android Studio คือโปรแกรมสำหรับพัฒนาแอป Android และมีเครื่องมือที่ช่วยจัดการสภาพแวดล้อมสำหรับการพัฒนา Android

Android SDK คืออะไร?

SDK ย่อมาจาก **Software Development Kit**

หมายถึง ชุดเครื่องมือสำหรับพัฒนาซอฟต์แวร์

ดังนั้น

Android SDK คือชุดเครื่องมือที่ใช้สำหรับพัฒนา ทดสอบ และ Build แอป Android

ในกระบวนการของเราจะเกี่ยวข้องกับสิ่งต่าง ๆ เช่น

- Android SDK
- Emulator
- เครื่องมือสำหรับ Build

Emulator คืออะไร?

Emulator คืออุปกรณ์จำลองที่ทำงานอยู่บนคอมพิวเตอร์

ในกรณีของ Android เราสามารถจำลองโทรศัพท์ Android บนคอมพิวเตอร์ เพื่อใช้ทดลองแอปได้

ตัวอย่างเช่น

คอมพิวเตอร์

↓

Android Emulator

↓

เปิดแอปที่เราสร้าง

จึงไม่จำเป็นต้องมีโทรศัพท์ Android จริงเพื่อทดลองทุกครั้ง

รายละเอียดเกี่ยวกับการเตรียม Environment และการ Build จะอยู่ในบทที่เกี่ยวข้องกับการ Build แอปภายหลัง

2.6 Xcode — เครื่องมือสำหรับ iPhone

ถ้าต้องการ Build แอปสำหรับ iPhone จะมีเครื่องมือของ Apple เข้ามาเกี่ยวข้อง โดยเฉพาะ **Xcode**

กระบวนการของ iOS มีข้อกำหนดเฉพาะของ Apple และแตกต่างจาก Android

ในช่วงต้นของหนังสือจึงยังไม่จำเป็นต้องลงรายละเอียดเกี่ยวกับ Xcode

เมื่อถึงขั้น Build และเตรียมแอปสำหรับเผยแพร่ เราจะกลับมาจัดการส่วนนี้โดยเฉพาะ

2.7 สรุปเครื่องมือที่ใช้ในหนังสือ

ตารางนี้คือสิ่งที่ควรจำจากบทนี้

เครื่องมือ	ใช้ทำอะไร
ChatGPT	ช่วยวางแผน สร้าง และแก้ Code
VS Code	เปิดและจัดการ Project และไฟล์ Code
Flutter	Framework สำหรับสร้างแอป
Firebase	Login, Database และระบบเบื้องหลัง
Android Studio	เครื่องมือสำหรับพัฒนาและจัดการ Android
Android SDK	ชุดเครื่องมือสำหรับพัฒนาและ Build Android
Xcode	เครื่องมือสำหรับพัฒนาและ Build ฝั่ง iPhone

จำหน้าที่ของแต่ละตัวให้ได้ก่อน ส่วนรายละเอียดจะค่อย ๆ เรียนรู้เมื่อถึงเวลาที่ต้องใช้งานจริง

2.8 สิ่งที่คุณจะได้ใช้จริงระหว่างสร้างแอป

เมื่อเริ่มลงมือจริง เครื่องมือเหล่านี้จะทำงานร่วมกัน

คุณ



กำหนดว่าอยากสร้างอะไร



ChatGPT

ช่วยวางแผนและสร้าง Code



VS Code

เปิด Project และจัดการไฟล์



Flutter

ทำหน้าที่เป็น Framework สำหรับสร้างแอป



Firebase

จัดการข้อมูลและระบบเบื้องหลังเมื่อแอปต้องใช้



Android / iPhone Tools

ใช้สำหรับทดสอบ Build และเตรียมเผยแพร่

ไม่จำเป็นต้องเข้าใจระบบทั้งหมดในครั้งเดียว

แต่เมื่อนั่งสื่อบอกให้คุณเปิดไฟล์ แก้ Code รันแอป หรือแก้ Error คุณ
จะรู้แล้วว่า กำลังใช้เครื่องมืออะไร และมันมีหน้าที่อะไร

บทที่ 3

เปลี่ยนไอเดียให้เป็นแผนแอป

คนส่วนใหญ่มักไม่ได้ติดตามการเขียน Code

แต่ติดตั้งแต่ก่อนเริ่มสร้างแอป

เพราะมีเพียงความคิดกว้าง ๆ เช่น

“อยากสร้างแอปรายรับรายจ่าย”

หรือ

“อยากทำแอปขายของ”

แต่ยังตอบไม่ได้ว่าแอปต้องมีอะไรบ้าง และผู้ใช้จะใช้งานอย่างไร

ถ้าเราให้ AI สร้างแอปจากคำอธิบายที่กว้างเกินไป AI ก็ต้องคาดเดา
รายละเอียดเอง และผลลัพธ์อาจไม่ตรงกับสิ่งที่เราต้องการ

ดังนั้นก่อนสร้างแอป เราจะเปลี่ยน

Idea → App Blueprint ก่อน

3.1 เริ่มจากปัญหา

อย่าเพิ่งเริ่มด้วย Feature

ให้เริ่มจากคำถามว่า

แอปนี้สร้างขึ้นมาเพื่อแก้ปัญหอะไร?

ตัวอย่าง

ยังไม่ชัด

“อยากสร้างแอปการเงิน”

ชัดขึ้น

“อยากให้คนที่ไม่เคยจดยรับรายจ่ายสามารถบันทึกเงินเข้าออกได้
ภายในไม่กี่วินาที”

ประโยคที่สองช่วยให้เราตัดสินใจเรื่อง Feature ได้ง่ายขึ้น เพราะเรารู้
แล้วว่าแอปกำลังพยายามแก้ปัญหอะไร

3.2 ระบุว่าใครคือผู้ใช้

ต่อไปให้ตอบว่า