

สร้างแอปอัจฉริยะ ด้วย CopilotKit

จากพื้นฐานสู่การใช้งานจริง

อนุชิต ชลอ

สำนักพิมพ์ก๊อปวาง



สร้างแอปอัจฉริยะด้วย CopilotKit

จากพื้นฐานสู่การใช้งานจริง

อนุชิต ชโลธร

คำนำสำนักพิมพ์

ในยุคที่เทคโนโลยีปัญญาประดิษฐ์ (AI) ก้าวหน้าไปอย่างก้าวกระโดด การสร้างสรรค์แอปพลิเคชันที่สามารถโต้ตอบ เรียนรู้ และทำงานร่วมกับมนุษย์ได้อย่างชาญฉลาด ไม่ใช่เรื่องไกลตัวอีกต่อไป แต่ได้กลายเป็นทักษะสำคัญที่นักพัฒนาซอฟต์แวร์ทุกคนควรมี

ในฐานะสำนักพิมพ์ที่มุ่งมั่นนำเสนอความรู้และเทคโนโลยีใหม่ๆ เราเล็งเห็นถึงความสำคัญของ CopilotKit ซึ่งเป็นชุดเครื่องมืออันทรงพลังที่จะช่วยให้นักพัฒนาสามารถผสมรวมความสามารถของ AI เข้ากับแอปพลิเคชันได้อย่างง่ายดายและมีประสิทธิภาพ CopilotKit ออกแบบมาเพื่อลดความซับซ้อนในการเชื่อมต่อกับโมเดล AI หลากหลายประเภท การจัดการการสนทนา การกำหนดฟังก์ชันการทำงานที่ AI สามารถเรียกใช้ (Actions) รวมถึงการให้บริบทที่จำเป็นเพื่อให้ AI ตอบสนองได้อย่างแม่นยำ

หนังสือ “สร้างแอปอัจฉริยะด้วย CopilotKit: จากพื้นฐานสู่การใช้งานจริง” เล่มนี้ จึงถือกำเนิดขึ้นด้วยความตั้งใจที่จะเป็นคู่มือที่สมบูรณ์แบบสำหรับนักพัฒนาทุกระดับ ตั้งแต่ผู้เริ่มต้นที่ต้องการทำความเข้าใจพื้นฐานของ CopilotKit ไปจนถึงผู้ที่มีประสบการณ์ที่ต้องการยกระดับแอปพลิเคชันของตนด้วยขีดความสามารถของ AI

ภายในเล่ม คุณจะได้เรียนรู้ตั้งแต่การติดตั้งและตั้งค่าโปรเจกต์ การสร้าง Chatbot ที่โต้ตอบได้ การออกแบบและเรียกใช้ Actions เพื่อให้ AI ทำงานตามคำสั่ง ไปจนถึงการประยุกต์ใช้ CopilotKit ในโปรเจกต์จริงที่หลากหลาย ด้วยตัวอย่างโค้ดที่ชัดเจน คำอธิบายที่เข้าใจง่าย และแนวทางการปฏิบัติที่สามารถนำไปใช้ได้จริง เราเชื่อมั่นว่าหนังสือเล่มนี้จะเป็นกุญแจสำคัญที่ช่วยให้คุณปลดล็อกศักยภาพของ AI และสร้างสรรค์นวัตกรรมใหม่ๆ ให้เกิดขึ้นได้

สำนักพิมพ์หวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นประโยชน์อย่างยิ่งต่อการเรียนรู้และพัฒนาทักษะของท่านผู้อ่านทุกท่าน ขอให้ทุกท่านสนุกกับการเดินทางในโลกของการพัฒนาแอปพลิเคชันอัจฉริยะ ด้วย CopilotKit ครับ

คำนำ

ในยุคที่ปัญญาประดิษฐ์ (AI) ไม่ใช่แค่แนวคิดในนิยายวิทยาศาสตร์อีกต่อไป แต่ได้กลายเป็นส่วนสำคัญที่ขับเคลื่อนนวัตกรรมและพลิกโฉมโลกดิจิทัล การสร้างสรรค์แอปพลิเคชันที่ชาญฉลาดและตอบสนองความต้องการของผู้ใช้งานได้อย่างลงตัวจึงไม่ใช่ทางเลือก แต่คือความจำเป็นสำหรับนักพัฒนาในปัจจุบัน

ตลอดหลายปีที่ผ่านมา การผสมรวมความสามารถของ AI โดยเฉพาะอย่างยิ่งโมเดลภาษาขนาดใหญ่ (LLMs) เข้ากับแอปพลิเคชันของเราแน่นแฟ้นไปด้วยความท้าทาย ทั้งเรื่องของโครงสร้างพื้นฐานที่ซับซ้อน การจัดการการเชื่อมต่อกับโมเดล AI ที่หลากหลาย การสร้างส่วนต่อประสานผู้ใช้ที่รองรับการโต้ตอบแบบสนทนา ไปจนถึงการออกแบบให้ AI สามารถ “เข้าใจ” และ “ดำเนินการ” ในสิ่งที่ผู้ใช้ต้องการได้อย่างแม่นยำ

นั่นคือจุดที่ CopilotKit ก้าวเข้ามา CopilotKit ไม่ใช่แค่ไลบรารีหรือเฟรมเวิร์กทั่วไป แต่มันคือชุดเครื่องมือที่ทรงพลัง ซึ่งออกแบบมาเพื่อลดความซับซ้อนของการพัฒนาแอปพลิเคชันอัจฉริยะ โดยเฉพาะไม่ว่าคุณจะต้องการสร้าง AI Assistant ที่สามารถโต้ตอบกับผู้ใช้ด้วยบทสนทนาที่ลื่นไหล, ระบบอัตโนมัติที่ AI สามารถเรียกใช้ฟังก์ชันการทำงานเฉพาะทางในแอปพลิเคชันของคุณ หรือแม้แต่การเพิ่มขีดความสามารถในการทำความเข้าใจบริบทของผู้ใช้ให้กับโมเดล AI ของคุณ

CopilotKit ทำให้กระบวนการเหล่านี้ง่ายขึ้นอย่างเหลือเชื่อ ช่วยให้
คุณมุ่งเน้นไปที่การสร้างคุณค่าและประสบการณ์ผู้ใช้ที่ยอดเยี่ยม
แทนที่จะต้องจมอยู่กับรายละเอียดทางเทคนิคที่ซับซ้อน

หนังสือเล่มนี้จะนำพาคุณดำดิ่งสู่โลกของ CopilotKit ตั้งแต่แนวคิด
พื้นฐาน การติดตั้ง และการตั้งค่าโปรเจกต์แรก ไปจนถึงเทคนิคขั้นสูง
ในการสร้างสรรค์ Actions และการจัดการ Context เพื่อให้ AI ของ
คุณทำงานได้อย่างชาญฉลาดและตอบโต้การใช้งานจริง เราจะ
สำรวจวิธีการเชื่อมต่อกับ AI Providers ยอดนิยม, การออกแบบ UI
สำหรับการโต้ตอบกับ AI และการประยุกต์ใช้ CopilotKit ใน
สถานการณ์ต่างๆ ผ่านตัวอย่างโค้ดและโปรเจกต์ที่จับต้องได้ เพื่อให้
คุณสามารถนำความรู้ที่ได้ไปต่อยอดในการพัฒนาแอปพลิเคชันของ
คุณเองได้ทันที

ไม่ว่าคุณจะเป็นนักพัฒนาที่คุ้นเคยกับ React/Next.js และ Node.js
ที่ต้องการยกระดับแอปพลิเคชันของคุณด้วย AI หรือเป็นผู้ที่กำลังมอง
หาเครื่องมือที่ช่วยให้การผสาน AI เข้ากับระบบทำได้ง่ายขึ้น หนังสือ
เล่มนี้คือคู่มือสำหรับคุณ ขอให้สนุกกับการเดินทางในโลกของการ
พัฒนาแอปพลิเคชันอัจฉริยะ ด้วย CopilotKit และสร้างสรรค์สิ่ง
ใหม่ๆ ที่เป็นประโยชน์ไปด้วยกัน

สารบัญ

ประวัติการพัฒนาแอปอัจฉริยะ	1
ทำไม AI ถึงกลายเป็นหัวใจสำคัญของแอปยุคใหม่?	1
โอกาสใหม่ที่เกิดจากแอปพลิเคชันอัจฉริยะ	2
ความท้าทายในการรวม AI เข้ากับแอปพลิเคชันจริง	4
CopilotKit สะพานเชื่อมนักพัฒนา กับพลังของ AI	5
รู้จักกับ CopilotKit	6
CopilotKit คืออะไร?	7
สถาปัตยกรรมและส่วนประกอบหลักของ CopilotKit	7
Frontend Integration (React/Next.js)	8
Backend Integration (Node.js/Next.js API Routes)	8
AI Model Integration	9
สรุปสถาปัตยกรรม	9
จุดเด่นของ CopilotKit	10
เปรียบเทียบ CopilotKit กับเครื่องมืออื่นในตลาด	12
LangChain / LlamaIndex	12
Native AI SDKs	13
ไลบรารีสำหรับสร้าง Chat UI	14
การทำงานร่วมกับ LLM และ Agent	14

เตรียมความพร้อมก่อนเริ่มใช้ CopilotKit	17
เครื่องมือที่แนะนำให้ติดตั้ง	18
Visual Studio Code (VS Code)	18
ติดตั้ง Node.js และ npm	19
สร้างโปรเจกต์ Next.js	19
ติดตั้ง CopilotKit	21
ตั้งค่า Environment Variables	22
การสร้างแอปพลิเคชัน CopilotKit	23
โครงสร้างโปรเจกต์เริ่มต้น	23
การตั้งค่า Backend สำหรับ CopilotKit	24
สร้างโฟลเดอร์สำหรับ API Route	25
สร้างไฟล์ route.ts	25
เขียนโค้ดใน route.ts	26
การตั้งค่า Frontend สำหรับ CopilotKit	27
แก้ไขไฟล์ layout.tsx (Root Layout)	27
เพิ่ม AI Assistant UI ใน page.tsx (Home Page)	28
การทดสอบแอปพลิเคชันพื้นฐาน	30
สร้าง Chat UI ด้วย CopilotKit	35
CopilotChat	35
CopilotSidebar	37
CopilotPopup	38

Headless UI	39
การปรับแต่ง UI	41
การปรับสีด้วย CSS Variables (วิธีที่ง่ายที่สุด)	41
การปรับแต่งด้วย Custom CSS	42
การปรับฟอนต์	45
การปรับไอคอน	46
การปรับข้อความ Label	47
เพิ่มความสามารถให้ Copilot สั่งการ UI ของคุณได้โดยตรง	49
useCopilotAction ให้ Copilot ลงมือทำใน Frontend	49
โครงสร้างการใช้งาน	49
ตัวอย่างการเพิ่มรายการ To-Do	50
การควบคุม Action	52
สร้าง Generative UI	53
การสร้างข้อความแนะนำอัตโนมัติด้วย	53
useCopilotChatSuggestions	
การทำงานร่วมกับ useCopilotReadable	54
การติดตั้งแพ็คเกจ	54
ตัวอย่างการใช้งานเบื้องต้น	54
การจัดการ Dependency	57
การใช้งาน Backend Actions และการเชื่อมต่อกับ Agents	59
การสร้าง Backend Actions ด้วย TypeScript (Node.js)	59

เพิ่ม Actions ลงใน CopilotRuntime	59
การสร้าง Backend Actions ด้วย Python	63
สร้าง FastAPI Server	64
สร้าง Backend Actions	64
การรัน Server	66
การเชื่อมต่อแอปกับ Remote Endpoint	66
ทดสอบการทำงาน	67
การปรับแต่ง Thread Pool Executor	68
การใช้ Generative UI ใน Copilot Chat	70
แสดงคอมโพเนนต์ UI แบบกำหนดเองในแชท	70
ตัวอย่างแสดงข้อมูลประชุมในปฏิทิน	70
ความหมายของสถานะ status	72
การดึงข้อมูลและแสดง UI ไปพร้อมกัน	73
ตัวอย่างแสดงการประชุมสุดท้ายของวัน	74
การใช้ Human-in-the-Loop (HITL)	76
ตัวอย่างยืนยันการประชุม	77
การแสดงความอย่างง่าย	79
ตัวอย่างสถานะการทำงาน	80
แสดง Action ที่ไม่รู้จักร่องหน้า (Catch All)	81
ตัวอย่างแสดงชื่อ Action ที่ถูกเรียก	82
การเชื่อมต่อข้อมูลของคุณเข้ากับ CopilotKit	84

การเชื่อมต่อข้อมูลฝั่งไคลเอนต์	84
การใช้ useCopilotReadable เพื่อแชร์ข้อมูลให้ Copilot	84
การเชื่อมต่อข้อมูลจากฝั่งเซิร์ฟเวอร์	87
การตั้งค่า Backend Action	88
จุดเด่นของการเชื่อมต่อกับ Backend	90
การเชื่อมต่อ MCP Server เข้ากับแอปของคุณ	92
ทำความรู้จักกับ MCP	92
เริ่มต้นใช้งาน MCP Server กับ CopilotKit Cloud	92
การใช้ MCP Server จาก Composio	93
ตั้งค่า CopilotKit Provider	93
เชื่อมต่อ MCP Server	94
เพิ่ม Copilot Chat UI	94
แสดงผลการเรียกใช้ MCP Tools แบบเรียลไทม์	95
รวมทุกอย่างเข้าด้วยกัน	96
การโฮสต์ MCP Server เอง	97
CopilotKit สำหรับ AI Agents	100
คุณสมบัติเด่นของ CoAgent	100
เริ่มต้นใช้งาน CopilotKit ร่วมกับ PydanticAI Agents	102
การสร้าง Agent ใหม่จากศูนย์	102
การเชื่อมต่อ Agent ที่มีอยู่แล้ว	103
สร้าง Frontend ด้วย Next.js	104

เชื่อมต่อ Frontend กับ Backend Agent	104
เพิ่ม CopilotKit ใน Layout ของแอป	105
แสดง CopilotKit UI สำหรับกล่องแชท	106
การใช้ Agentic Chat UI	107
การใช้ CopilotChat	108
การใช้ CopilotSidebar	109
การใช้ CopilotPopup	111
การใช้ Headless UI	112
การใช้ Generative UI ใน CoAgent	114
วิธีการใช้งาน Generative UI	114
Agentic Generative UI	115
การเชื่อมต่อ Agent	115
จำลองการอัปเดตสถานะ	116
แสดงสถานะของ Agent	116
แสดงสถานะนอกแชต	117
Tool-based Generative UI	118
การสร้าง Agent และเพิ่ม Tool	119
การแสดงผล Tool Call ใน Frontend	119
Human in the Loop (HITL) ใน CoAgent	121
Flow-based Agent คืออะไร?	121
วิธีการใช้งาน Flow-based Agent	122

สร้างและเชื่อมต่อ Agent กับ CopilotKit	122
ติดตั้ง CopilotKit SDK	122
เพิ่ม useCopilotAction ใน Frontend	122
ตั้งค่า Agent ด้วย PydanticAI	124
การ Shared State ระหว่าง UI และ Agent	126
Shared State คืออะไร?	126
การอ่านสถานะของ Agent แบบเรียลไทม์	127
กำหนดสถานะของ Agent	127
การใช้งาน Hook useCoAgent	128
การแสดงผลสถานะของ Agent ภายในหน้าต่างแชต	129
การเขียนค่าลงในสถานะของ Agent	131
การเรียกใช้งาน Agent และเชื่อมต่อกับ CopilotKit	131
การกำหนดสถานะของ Agent	131
การเรียกใช้ฟังก์ชัน setState จาก Hook useCoAgent	133
การเรียกให้ Agent ทำงานใหม่	134
การอัปเดตสถานะล่วงหน้า	136
วิธีการอัปเดตสถานะล่วงหน้า	137
CopilotKit สำหรับ ADK	145
สร้างโปรเจกต์ CopilotKit พร้อม ADK Agent	145
สร้างโปรเจกต์ใหม่	145
ติดตั้ง dependency	146

ตั้งค่า Google API Key	146
รัน Agent	146
เชื่อมต่อ CopilotKit กับ ADK Agent ที่มีอยู่แล้ว	147
ติดตั้ง dependency สำหรับ Agent	147
แก้ไขไฟล์ Agent	148
สร้าง Frontend	149
ติดตั้ง dependency สำหรับ CopilotKit	149
สร้าง Route สำหรับ CopilotRuntime	150
ตั้งค่า CopilotKit Provider	151
เรียกใช้ CopilotKit UI	152
รัน Agent	153
ตัวอย่างโปรเจกต์	155
แอปพลิเคชัน Todo แบบ Direct LLM	155
แอปสรุปข้อความจากเว็บไซต์ แบบ CoAgent	173
ดาวน์โหลดซอร์สโค้ด	187

ปฏิวัติการพัฒนาแอปอัจฉริยะ

ในยุคที่เทคโนโลยีดิจิทัลขับเคลื่อนแทบทุกมิติของชีวิต แอปพลิเคชันที่เราเคยใช้เพียงเพื่อดูข้อมูลหรือทำงานพื้นฐาน กำลังเปลี่ยนโฉมหน้ากลายเป็น **“ผู้ช่วยส่วนตัวอัจฉริยะ”** ที่เข้าใจเราอย่างลึกซึ้ง ตอบโจทย์เฉพาะบุคคล และเรียนรู้ได้จากประสบการณ์ของเราโดยตรง

หัวใจของการเปลี่ยนแปลงนี้คือ **ปัญญาประดิษฐ์ (AI)** ซึ่งไม่ได้เป็นเพียงฟีเจอร์เสริมอีกต่อไป แต่กลายเป็น “รากฐาน” ใหม่ของการออกแบบแอปยุคถัดไป บทนี้จะพาคุณเข้าใจว่า AI เปลี่ยนโลกของนักพัฒนาอย่างไร โอกาสที่เปิดกว้างขึ้นมีอะไรบ้าง และเราจะรับมือกับความท้าทายที่มาพร้อมกับมันได้อย่างไร

ทำไม AI ถึงกลายเป็นหัวใจสำคัญของแอปยุคใหม่?

เมื่อก่อน แอปพลิเคชันทำงานตามเงื่อนไขที่นักพัฒนามาหนดไว้ตายตัว เช่น ถ้าผู้ใช้กดปุ่ม A ก็แสดงข้อความ B แต่โลกจริงซับซ้อนกว่านั้นมาก ผู้ใช้แต่ละคนมีพฤติกรรม ความต้องการ และบริบทที่แตกต่างกัน จึงไม่แปลกที่แอปแบบเก่าจะตอบสนองความต้องการยุคใหม่ได้น้อยลง

AI เข้ามาช่วยในจุดนี้ เพราะมันมีคุณสมบัติที่เหนือกว่าการเขียนกฎธรรมดา

- **เรียนรู้และพัฒนาได้ด้วยตัวเอง** AI ไม่ได้รอคำสั่ง แต่สามารถเรียนรู้จากข้อมูล พฤติกรรม และสถานการณ์ แล้วนำไปปรับใช้ในการตัดสินใจครั้งถัดไป
- **ประมวลผลข้อมูลขนาดใหญ่แบบเรียลไทม์** ขณะที่มนุษย์อาจต้องใช้เวลาหลายชั่วโมงในการวิเคราะห์ข้อมูล AI สามารถสแกนวิเคราะห์ และสรุปข้อมูลระดับ Big Data ได้ภายในเสี้ยววินาที
- **ปรับเปลี่ยนตามผู้ใช้แต่ละคน** แอปที่มี AI สามารถรู้ได้ว่าผู้ใช้งานชอบอะไร เคยดูอะไร หรือมีแนวโน้มจะสนใจอะไร แล้วปรับการแสดงผลให้ตรงใจโดยอัตโนมัติ
- **ลดความผิดพลาดและทำงานซ้ำๆ แทนมนุษย์ได้** ไม่ว่าจะเป็นการตอบอีเมล ตรวจสอบคุณภาพ หรือทำรายงานประจำวัน AI สามารถช่วยแบ่งเบาภาระงานที่ต้องการความแม่นยำสูงได้อย่างมีประสิทธิภาพ
- **โต้ตอบกับผู้ใช้ได้ด้วยภาษาธรรมชาติ** เทคโนโลยี NLP และ LLMs อย่าง ChatGPT ทำให้แอปสามารถสนทนา ตอบคำถาม หรือให้คำแนะนำผู้ใช้ด้วย “ภาษาคน” ได้อย่างสิ้นไหล

โอกาสใหม่ที่เกิดจากแอปพลิเคชันอัจฉริยะ

การรวม AI เข้ากับแอปพลิเคชันไม่ได้เป็นเพียง “ของเล่นล้ำสมัย” แต่เป็นการเปิดประตูสู่โอกาสใหม่ที่ไม่เคยเกิดขึ้นมาก่อน

- **AI Assistant ที่เข้าใจผู้ใช้จริงๆ** จาก Chatbot ธรรมดา กลายเป็นผู้ช่วยอัจฉริยะที่จำชื่อผู้ใช้ จำบริบทได้ และสามารถช่วยจัดการงานต่างๆ ได้หลากหลาย ตั้งแต่การจองร้านอาหารไปจนถึงวางแผนชีวิต
- **การสร้างเนื้อหาอัตโนมัติ (Generative AI)** ด้วยโมเดล LLMs ปัจจุบัน แอปสามารถ “คิด” และ “สร้าง” เนื้อหาใหม่ได้เอง เช่น เขียนบทความ เขียนโค้ด สร้างภาพ หรือแต่งเพลง ช่วยเร่งงานสร้างสรรค์ได้แบบก้าวกระโดด
- **การวิเคราะห์เพื่อคาดการณ์อนาคต** ไม่ใช่แค่อ่านข้อมูลในอดีต แต่ AI สามารถคาดการณ์แนวโน้มล่วงหน้า เช่น พฤติกรรมผู้ใช้ การขาย การเงิน หรือความเสี่ยง
- **ระบบแนะนำแบบเฉพาะบุคคล (Smart Recommendation)** แนะนำคอนเทนต์หรือสินค้าแบบที่ผู้ใช้ไม่เคยคาดคิดมาก่อน แต่กลับ “ใช่” อย่างน่าประหลาดใจ
- **เครื่องมือเพิ่มประสิทธิภาพ (Productivity Tools)** AI ถูกผสานเข้ากับเครื่องมือในชีวิตประจำวัน เช่น ช่วยสรุปประชุม เขียนโค้ด ตรวจไวยากรณ์ หรือจัดลำดับความสำคัญของงาน
- **อุตสาหกรรมเฉพาะทางก็ได้ประโยชน์** เช่น ในวงการแพทย์ AI ช่วยวินิจฉัยโรคแม่นยำยิ่งขึ้น หรือในโรงงาน AI ช่วยตรวจสอบ

คุณภาพสินค้าอัตโนมัติและทำนายปัญหาล่วงหน้า

ความท้าทายในการรวม AI เข้ากับแอปพลิเคชันจริง

ถึงแม้ AI จะทรงพลัง แต่การผสานเข้ากับแอปจริงๆ กลับไม่ง่าย โดยเฉพาะสำหรับนักพัฒนาที่ต้องรับมือกับความซับซ้อนเบื้องหลัง

- **โค้ดซับซ้อนเกินไป** AI APIs โดยเฉพาะ LLMs มักต้องจัดการสิ่งต่างๆ เช่น Prompt, Token, Context, Memory ซึ่งไม่ง่ายสำหรับมือใหม่
- **เลือกโมเดลให้เหมาะกับงาน** ปัจจุบันมีโมเดลมากมาย เช่น GPT-4, Claude, Gemini และ LLaMA การเลือกให้ตรงกับ Use Case และบริหารต้นทุนให้เหมาะสมเป็นความท้าทาย
- **จัดการบริบท (Context) ของผู้ใช้** จะส่งข้อมูลอะไรให้ AI เพื่อให้เข้าใจสถานการณ์ได้โดยไม่ทำให้แอปช้าหรือซับซ้อนเกินไป?
- **ทำให้ AI “ลงมือทำ” ได้จริง** แค่ตอบข้อความไม่พอ แอปยุคใหม่ต้องให้ AI เรียก API ได้ ทำงานจริงได้ เช่น เพิ่มข้อมูลในฐานข้อมูล ส่งอีเมล หรือเชื่อมต่อระบบหลังบ้าน
- **ปัญหาเรื่อง Latency** โมเดล AI ใหญ่ๆ มักมีความหน่วงเวลา การออกแบบแอปให้ยังลื่นไหลแม้ต้องรอผลลัพธ์จาก AI เป็นเรื่องที่ต้องวางแผนให้ดี
- **UX ที่เหมาะกับ AI** ผู้ใช้ไม่อยากรอ ไม่อยากพิมพ์ยาวๆ หรือสับสนว่า AI จะทำอะไรได้บ้าง – การออกแบบ UI/UX ให้ AI เป็น

“ผู้ช่วย” ที่เป็นธรรมชาติคือกุญแจสำคัญ

- **ต้นทุนสูง** เรียกใช้ AI โดยเฉพาะ LLMs ผ่าน API จากผู้ให้บริการใหญ่มักมีค่าใช้จ่ายสูง จึงต้องมีการจัดการอย่างมีประสิทธิภาพ เช่น caching หรือ fallback ไปใช้โมเดลเล็ก
- **จริยธรรมและความปลอดภัย** AI สามารถใช้ข้อมูลผิดวัตถุประสงค์ หรือให้คำแนะนำที่ไม่เหมาะสมได้ จึงต้องออกแบบระบบให้ปลอดภัย โปร่งใส และตรวจสอบได้

CopilotKit สะพานเชื่อมนักพัฒนา กับพลังของ AI

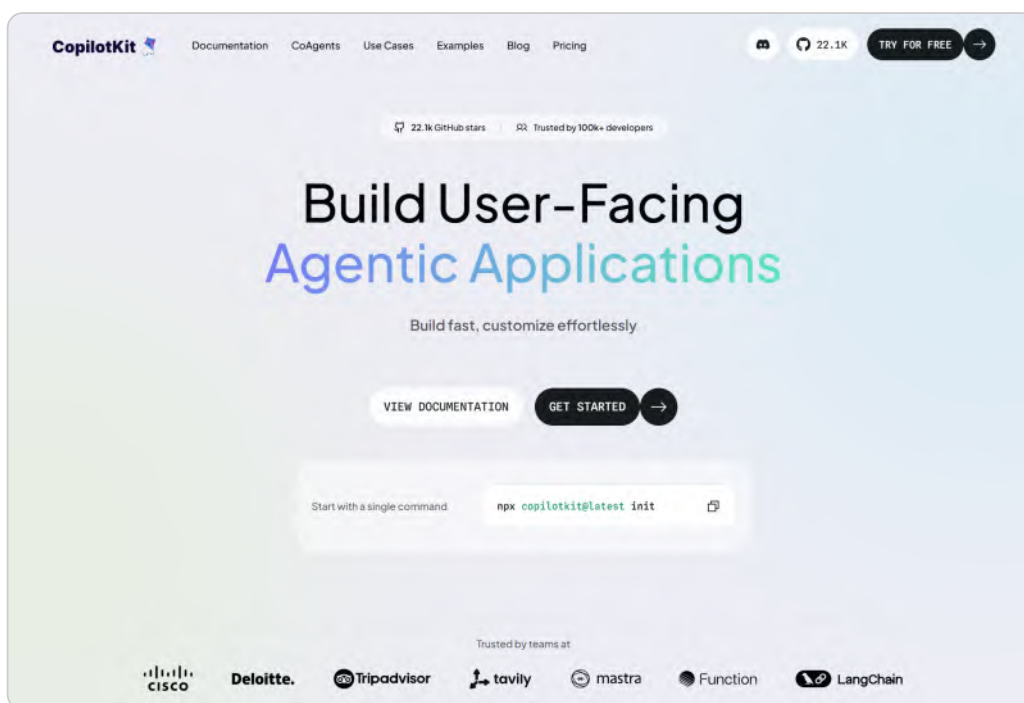
ด้วยความซับซ้อนเหล่านี้ เครื่องมือใหม่อย่าง **CopilotKit** จึงถือกำเนิดขึ้นเพื่อช่วยให้นักพัฒนาไม่ต้องเริ่มจากศูนย์ ไม่ต้องจัดการทุกอย่างเอง แต่สามารถสร้าง AI ที่ใช้งานได้จริง รวดเร็ว และปลอดภัย

CopilotKit ถูกออกแบบมาเพื่อให้คุณสามารถสร้างแอปที่ “มีผู้ช่วยอัจฉริยะ” ได้ในเวลาไม่กี่นาที รองรับทั้งฝั่ง Frontend (React/Next.js) และ Backend (Node.js/Express) พร้อมระบบ Action/Tool ที่ให้ AI เชื่อมต่อกับโลกจริงได้ทันที

ในบทถัดไป เราจะพาคุณเจาะลึกเข้าไปใน CopilotKit — ตั้งแต่วิธีติดตั้ง การสร้าง AI Assistant จนถึงการ **เขียนแอปเชื่อมต่อกับ LLM โดยตรง** โดยใช้ OpenAI หรือโมเดลที่รันในเครื่องของคุณเอง

รู้จักกับ CopilotKit

หลังจากที่เราได้ทำความเข้าใจถึงบทบาทของ AI ในการพัฒนาแอปพลิเคชันยุคใหม่ และเรียนรู้ความท้าทายต่างๆ ที่นักพัฒนาต้องเผชิญในการนำ AI เข้ามาผนวกใช้ในระบบงานจริง บทนี้จะพาคุณเข้าสู่หัวใจสำคัญของหนังสือเล่มนี้ นั่นก็คือ **CopilotKit** — เครื่องมือที่ออกแบบมาเพื่อให้ นักพัฒนาสามารถสร้างแอปพลิเคชันที่ขับเคลื่อนด้วย AI ได้อย่างมีประสิทธิภาพและง่ายดาย



เว็บไซต์ CopilotKit

เนื้อหาในบทนี้จะพาคุณไปรู้จักกับแนวคิดเบื้องหลังการพัฒนา CopilotKit อธิบายสถาปัตยกรรมโดยรวม จุดเด่นต่างๆ ของเครื่อง