



เร่งสปีดการพัฒนาซอฟต์แวร์ ด้วยพลัง AI

อนุชิต ชลธร

สำนักพิมพ์ก๊อปวาง

สูตรลับ Gemini CLI

เร่งสปีดการพัฒนาซอฟต์แวร์ด้วยพลัง AI

อนุชิต ชโลธร

(อัปเดตครั้งที่ 2)

คำนำสำนักพิมพ์

ในนามของสำนักพิมพ์ เรารู้สึกยินดีอย่างยิ่งที่ได้นำเสนอหนังสือ

“สูตรลับ Gemini CLI: เร่งสปีดการพัฒนาซอฟต์แวร์ด้วยพลัง AI”

สู่สายตานักพัฒนาชาวไทยทุกท่าน

โลกการพัฒนาซอฟต์แวร์ไม่เคยหยุดนิ่ง การเปลี่ยนแปลงคือสิ่งเดียวที่คงอยู่ และการมาถึงของปัญญาประดิษฐ์เชิงสนทนา

(Conversational AI) คือหนึ่งในก้าวกระโดดครั้งสำคัญที่สุดในรอบทศวรรษ AI ได้พลิกโฉมแนวคิดเรื่องประสิทธิภาพการทำงาน และเปิดประตูสู่ความเป็นไปได้ใหม่ๆ ที่เราแทบไม่เคยนึกถึง

หนังสือเล่มนี้ไม่ใช่เพียงคู่มือการใช้เครื่องมือ แต่คือบันทึกการเดินทางสู่ยุคใหม่ของการเขียนโค้ด ยุคที่นักพัฒนาสามารถทำงานเคียงข้าง AI ได้อย่างราบรื่นผ่าน Command-Line Interface (CLI) อันเป็นพื้นที่ทำงานที่คุ้นเคย เนื้อหาในเล่มสกัดจากประสบการณ์ตรง แสดงให้เห็นว่า Gemini CLI สามารถเป็นทั้งผู้ช่วย, ที่ปรึกษา และเพื่อนร่วมทีมที่พร้อมช่วยคุณแก้ปัญหา ตั้งแต่การสร้างโค้ดขึ้นเล็ก ไปจนถึงการออกแบบโครงสร้างโปรเจกต์ที่ซับซ้อน

เรามั่นใจว่าทักษะการใช้เครื่องมือ AI จะกลายเป็นหนึ่งในพื้นฐานสำคัญของนักพัฒนาในอนาคต และหวังว่าหนังสือเล่มนี้จะเป็นจุดเริ่ม

ต้นที่มั่นคงและเป็นแรงบันดาลใจให้คุณนำเทคโนโลยีนี้ไปต่อยอด
สร้างผลงานที่ยิ่งใหญ่ต่อไป

ขอให้ทุกท่านเพลิดเพลินกับการเรียนรู้ และค้นพบศักยภาพใหม่ๆ ใน
การทำงานของคุณ

คำนำผู้เขียน

ในยุคที่ปัญญาประดิษฐ์ (AI) เข้ามามีบทบาทในชีวิตประจำวันอย่างรวดเร็ว วงการพัฒนาซอฟต์แวร์ก็เป็นหนึ่งในสาขาที่ได้รับการพลิกโฉมอย่างชัดเจน เครื่องมือขับเคลื่อนด้วย AI ได้ก้าวจากโลกนิยายวิทยาศาสตร์ มาสู่การเป็นผู้ช่วยทรงพลังในมือของนักพัฒนาทั่วโลก

หนังสือ **“สูตรลับ Gemini CLI: เร่งสปีดการพัฒนาซอฟต์แวร์ด้วยพลัง AI”** เล่มนี้ จัดทำขึ้นเพื่อเป็นคู่มือสำหรับนักพัฒนาทุกระดับ ที่ต้องการนำศักยภาพของ Large Language Models (LLMs) มาประยุกต์ใช้กับการทำงาน ผ่านเครื่องมือที่เรียบง่ายแต่ทรงพลังอย่าง Gemini CLI เราจะพาคุณสำรวจว่าเครื่องมือนี้สามารถยกระดับการเขียนโค้ด การแก้บั๊ก และการสร้างสรรค์โปรเจกต์ได้อย่างไร

เรามองว่า AI ไม่ได้มาเพื่อแทนที่นักพัฒนา แต่เป็นเครื่องมือที่ช่วยเสริมพลัง ลดภาระงานที่ซ้ำซาก และเปิดโอกาสให้เรามุ่งไปที่การแก้ปัญหาที่ซับซ้อนและการสร้างนวัตกรรมใหม่ๆ Gemini CLI คือสะพานเชื่อมระหว่างความสามารถของโมเดลภาษากับเวิร์กโฟลว์การทำงานบนเทอร์มินัลที่คุณคุ้นเคย

ไม่ว่าคุณจะเป็นนักพัฒนามือใหม่ที่กำลังมองหาเครื่องมือทันสมัย หรือเป็นนักพัฒนามากประสบการณ์ที่ต้องการเพิ่มประสิทธิภาพ

หนังสือเล่มนี้ได้รวบรวมเทคนิค ตัวอย่าง และกรณีศึกษาที่จะช่วยให้คุณเริ่มต้นได้ทันที และนำไปประยุกต์ใช้ได้จริง

ขอให้คุณเพลิดเพลินกับการเดินทางสู่โลกการพัฒนาซอฟต์แวร์ยุคใหม่ ที่ซึ่งความคิดสร้างสรรค์ของคุณและพลังของ AI จะทำงานร่วมกัน อย่างไร้ขีดจำกัด

สารบัญ

เริ่มต้นใช้งาน Gemini CLI	1
Gemini CLI คืออะไร	1
ทำไมต้องใช้ Gemini CLI ในงานพัฒนา	2
การติดตั้งและตั้งค่า	3
ติดตั้งผ่าน npx สำหรับผู้ใช้ Node.js	3
ติดตั้งผ่าน Homebrew	3
การยืนยันตัวตน	4
วิธีที่ 1: Login with Google (แนะนำ)	4
วิธีที่ 2: ใช้ Gemini API Key	4
วิธีที่ 3: ใช้ Vertex AI (สำหรับองค์กร)	5
การตั้งค่า Environment Variables แบบถาวร	6
ยืนยันการติดตั้งและคำสั่งเบื้องต้น	7
เริ่มต้นการสนทนา	7
ใช้ Pipe ส่งต่อข้อมูล	8
การใช้งานพื้นฐาน Gemini CLI สำหรับนักพัฒนา	10
คำสั่งพื้นฐานและโครงสร้างการใช้งาน	10
การสร้างข้อความแบบโต้ตอบ	11
ตัวอย่างการถามคำถามทั่วไป	11

ตัวอย่างการสร้างเนื้อหา	11
การทำงานกับโปรเจกต์	11
การกำหนดโมเดลและพารามิเตอร์	12
การเรียกใช้คำสั่งพิเศษ	13
การให้บริบทกับโปรเจกต์ด้วย GEMINI.md	13
การสลับโหมดแชตและโหมดเทอร์มินัล	14
การเชื่อมต่อกับ MCP Server	14
Gemini CLI กับการสร้างโค้ดอัตโนมัติ	16
การสร้างฟังก์ชันและสคริปต์	16
ตัวอย่าง Python	16
การสร้างคอมโพเนนต์ UI (HTML, CSS, React)	17
ตัวอย่าง React & Tailwind CSS	17
การสร้าง Unit Test และ Test Cases	20
ตัวอย่าง JavaScript - Jest	20
การประยุกต์ใช้กับภาษาโปรแกรมมอดินิยม	22
การปรับปรุงและ Refactor โค้ด	24
การ Refactor โค้ดที่ซับซ้อนให้เข้าใจง่าย	24
ตัวอย่าง JavaScript	24
การแปลโค้ดข้ามภาษาโปรแกรม	26
ตัวอย่าง แปลงโค้ด Python ไปเป็น Go	27
การอัปเดตเวอร์ชันไลบรารีและเฟรมเวิร์ก	27

ตัวอย่าง Express.js	28
การหาจุดที่ควรปรับปรุงประสิทธิภาพ (Code Optimization)	29
ตัวอย่าง Python	29
การแก้ไขข้อบกพร่อง (Debugging)	31
การอธิบายข้อความแสดงข้อผิดพลาด (Error Messages)	31
ตัวอย่าง Python	31
การสร้างโค้ดเพื่อแก้ไขบั๊ก	33
ตัวอย่าง JavaScript	33
การค้นหาข้อผิดพลาดทางตรรกะ (Logic Errors)	34
ตัวอย่าง Logic Bug	35
เครื่องมือในตัวของ Gemini CLI (Built-in Tools)	38
การทำงานกับไฟล์ (File System Operations)	38
การอ่านและสรุปเนื้อหาไฟล์	38
การค้นหาข้อมูลภายในโปรเจกต์	39
การสร้างและแก้ไขไฟล์	39
การสร้างไฟล์ GEMINI.md เพื่อให้บริบท	39
การรันคำสั่ง Shell (Shell Commands)	40
ตัวอย่างการใช้งาน	40
การเข้าถึงข้อมูลบนเว็บ (Web Fetch & Search)	41
การดึงข้อมูลจาก URL โดยตรง (Web Fetch)	41
การค้นหาด้วย Google Search	41

การใช้งาน Gemini CLI ร่วมกับ Workflow ของคุณ	43
การสร้าง Alias สำหรับคำสั่งที่ใช้บ่อย	43
ตัวอย่างสำหรับ Bash/Zsh	43
การใช้งานร่วมกับ Git (เช่น สร้าง Commit Message)	44
ตัวอย่างคำสั่ง (สามารถสร้างเป็น Alias ได้)	45
การประยุกต์ใช้ใน CI/CD Pipeline	45
ตัวอย่างสคริปต์ใน CI/CD	46
การสร้างเครื่องมือต่อยอด (Custom Tools)	47
ตัวอย่างสคริปต์สร้างคอมโพเนนต์ React อัตโนมัติ	47
ตัวอย่างโปรเจกต์	49
สร้าง Full-Stack To-Do Application	49
ภาพรวมโครงการ	50
เทคโนโลยีที่ใช้	50
ไฟล์กำหนดรายละเอียดโครงการ	51
วิเคราะห์ข้อมูลด้วย Pandas	56
ภาพรวมโครงการ	56
โครงสร้างโปรเจกต์	57
ไฟล์กำหนดรายละเอียดโครงการ	58
สร้าง Mobile App ด้วย Flutter เชื่อมต่อกับ Firebase	61
เทคโนโลยีที่ใช้	61
โครงสร้างข้อมูล	61

พีเจอรหลักของแอป	62
การใช้งานอย่างปลอดภัยและข้อควรระวัง	68
ปฏิบัติตามหลักการ “สิทธิ์น้อยที่สุด” (Principle of Least Privilege)	68
ตรวจสอบทุกโค้ดที่ AI สร้างขึ้นเสมอ	68
การจัดการข้อมูลและข้อมูลลับ (Data and Secrets)	69
ความเสี่ยงจากการรันคำสั่ง (Command Execution)	70
ตรวจสอบ Dependency และ Supply Chain	70
สรุป checklist เพื่อความปลอดภัย	71
การสร้างคำสั่งส่วนตัวและส่วนขยาย (Custom Commands)	72
คำสั่งอย่างง่ายด้วยไฟล์ TOML	72
หลักการทำงาน	72
โครงสร้างของไฟล์ TOML	73
ตัวอย่าง สร้างคำสั่งสำหรับเขียน Unit Test	74
ส่วนขยายขั้นสูงด้วย MCP Servers	75
ขั้นตอนการทำงาน (ภาพรวม)	75
เทคนิคการเขียน Prompt ที่ดี	77
มีความชัดเจนและเจาะจง (Be Clear and Specific)	77
ให้บริบทที่เพียงพอ (Provide Sufficient Context)	78
กำหนดบทบาท (Assign a Role)	79

ใช้ตัวอย่างประกอบ (Use Examples - Few-shot Prompting)	79
แบ่งงานที่ซับซ้อนออกเป็นขั้นตอนย่อยๆ (Break Down Complex Tasks)	80
บอกให้คิดเป็นขั้นเป็นตอน (Chain of Thought Prompting)	81
ขยายความสามารถด้วย MCP Servers	83
MCP และไลบรารี FastMCP	83
ตัวอย่าง: สร้างเครื่องมือสภาพอากาศด้วย FastMCP	83
เตรียมโปรเจกต์ (Python)	84
สร้างไฟล์เครื่องมือ	84
การติดตั้ง	86
การเรียกใช้งาน	87
ส่วนขยาย (Extension) ใน Gemini CLI	88
การจัดการส่วนขยาย	88
แสดงรายการส่วนขยายที่ติดตั้ง	88
การติดตั้งส่วนขยาย	88
การถอดการติดตั้งส่วนขยาย	89
การปิดใช้งานส่วนขยาย	89
การเปิดใช้งานส่วนขยาย	90
การอัปเดตส่วนขยาย	90
รายชื่อ Extensions	91

การสร้างส่วนขยาย	98
สร้าง Extension พื้นฐานจาก Boilerplate	98
การเชื่อมต่อ Extension ในเครื่อง	98
การทำงานของระบบ	99
การใช้ตัวแปร (Variables)	102
การทำงานร่วมกับ GitHub	104
Gemini CLI GitHub Action คืออะไร	104
ประโยชน์และการนำไปใช้	104
การรีวิวกัดอัตโนมัติ (Automated Code Reviews)	104
การจัดการ Issues อัตโนมัติ (Issue Triage)	105
การขอความช่วยเหลือตามต้องการ (On-demand Assistance)	106
การตั้งค่าเบื้องต้น	106
Headless Mode และการใช้งานในสคริปต์	108
Headless Mode คืออะไร	108
วิธีใช้งาน	108
การรับผลลัพธ์แบบข้อความ (Text)	108
การรับผลลัพธ์แบบ JSON (Structured Output)	109
กรณีการใช้งาน	112
การทำงานร่วมกับ IDE	113
การใช้งาน Gemini CLI ใน VS Code	113

เปิด Integrated Terminal	113
เริ่มต้นใช้งาน	113
เวิร์กโฟลว์ที่มีประสิทธิภาพใน VS Code	114
การสร้างโค้ดจากไฟล์ที่เปิดอยู่	115
การดีบั๊กและแก้ปัญหา	115
ส่วนขยาย Gemini CLI Companion สำหรับ VS Code	116
คุณสมบัติหลัก	117
การใช้งานร่วมกับ VS Code	118
ประสบการณ์การพัฒนาแบบไร้รอยต่อ	119
Sandboxing และความปลอดภัย	120
โมเดลความปลอดภัย: คุณคือผู้ควบคุม	120
การขอคำยืนยัน (Confirmation Prompt)	120
การจำกัดขอบเขตการทำงาน (Sandboxing)	121
Sandbox ทำงานอย่างไร	121
การเปิดใช้งาน Sandbox	122
แนวทางปฏิบัติเพื่อความปลอดภัยสูงสุด	122
Trusted Folders เพื่อเวิร์กโฟลว์ที่ลื่นไหล	124
Trusted Folders คืออะไร	124
การตั้งค่า Trusted Folders	124
ขั้นตอนการตั้งค่า	125
คำสั่งประเภทไหนที่ไม่ต้องยืนยัน	126

ข้อควรระวังและแนวทางปฏิบัติที่ดีที่สุด	126
คำสั่งที่ใช้บ่อย	128
หมวดที่ 1: การโต้ตอบ (Interaction)	128
หมวดที่ 2: การทำงานกับไฟล์ (File Operations)	129
หมวดที่ 3: การตั้งค่า (Configuration)	129
ทิปและเทคนิคการสร้างไฟล์ GEMINI.md	131
เริ่มต้นง่ายๆ ไม่ต้องสมบูรณ์แบบ	131
ใช้ Markdown เพื่อจัดระเบียบ	131
ระบุเป้าหมายปัจจุบัน	132
กำหนดรายการ Tasks ด้วย Checklist	132
เป็น “พิมพ์เขียว” ไม่ใช่ “สมุดจด”	134
อัปเดตให้ทันสมัยอยู่เสมอ	135
ใช้สำหรับงานที่ซับซ้อนและหลายไฟล์	135
ดาว์นโหลดซอร์สโค้ด	136