

สร้างแอปพลิเคชันแบบมือโปร ด้วย

Next.js & ChatGPT

- การใช้ Next.js สร้างแอปพลิเคชันทั้งในฝั่ง Frontend และ Backend
- เทคนิคการนำ ChatGPT และ ChatGPT Plus มาช่วยพัฒนาโปรแกรม
- การใช้ Next.js ร่วมกับฐานข้อมูล MongoDB และ Firebase Firestore Database
- สร้างระบบยืนยันตัวตนโดยใช้ Next-Auth และ Firebase Authentication
- การใช้ Tailwind CSS ปรับแต่งแอปพลิเคชันในแบบ Responsive Design
- อธิบายครบทุกเรื่องเกี่ยวกับ Next.js เช่น Server Actions, Middleware และอื่น ๆ

จิราวุธ วารินทร์

SCAN FREE



ไฟล์ประกอบหนังสือ

สารบัญ

Part 1 Basic React for Next.js		
บทที่ 1	แนะนำ Next.js	6
บทที่ 2	คอมโพเนนต์และหน้าเว็บเพจ	25
บทที่ 3	การกำหนดสไตล์และการใช้งาน Tailwind CSS	42
บทที่ 4	การใช้งาน props	72
บทที่ 5	พื้นฐานการใช้งาน state	88
บทที่ 6	การใช้ฮุคในฟังก์ชันคอมโพเนนต์	111
บทที่ 7	พื้นฐานการใช้งาน Context API	141
Part 2 Essential Next.js		
บทที่ 8	การใช้งานเร้าต์ติ้งใน Next.js	166
บทที่ 9	การกำหนด Layout และเมนูใน Next.js	195
บทที่ 10	วิธี fetch ข้อมูลมาใช้ในคอมโพเนนต์	231
บทที่ 11	แบบฟอร์มและการใช้งาน Server Actions	260
บทที่ 12	Metadata และ Image	288
บทที่ 13	Next.js กับฐานข้อมูล MongoDB	300
บทที่ 14	Route Handler ใน Next.js	347
บทที่ 15	ตัวอย่างการสร้างและใช้งาน API	376
บทที่ 16	การใช้งาน Middleware ใน Next.js	409
บทที่ 17	ยืนยันตัวตนโดยใช้ Next-Auth	426
Part 3 Create Project with ChatGPT		
บทที่ 18	ChatGPT กับการพัฒนาโปรแกรม	454
บทที่ 19	เรียนรู้ Firebase ด้วย ChatGPT	485
บทที่ 20	การสั่ง Build และ Deploy แอปพลิเคชัน	534

Part 1

Basic React
for Next.js

01 | แนะนำ Next.js

Next.js คือ เฟรมเวิร์คที่ช่วยให้เราสามารถสร้างแอปพลิเคชันด้วย React ทำได้ง่ายและรวดเร็วยิ่งขึ้น โดย Next.js ถูกสร้างมาเพื่อรองรับการใช้งานจริง (Productions) โดยเฉพาะแอปพลิเคชันที่มีผู้ใช้งานจำนวนมาก

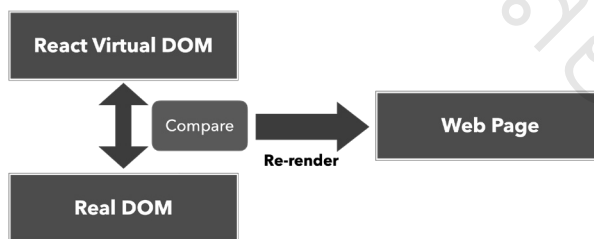
Next.js ได้เตรียมคุณสมบัติหลายอย่างซึ่งช่วยให้การโหลดเว็บเพจทำได้เร็ว โดยสามารถเรนเดอร์เนื้อหาบางส่วนเตรียมไว้ก่อน (Pre-rendering) และมีระบบการจัดเก็บข้อมูลลงในแคช ทำให้หน้าเว็บเพจสามารถแสดงผลได้สมบูรณ์ตั้งแต่การโหลดใช้งานในครั้งแรก (คุณสมบัติ Server-Side Rendering และ Static Site Generation) ด้วยการที่เว็บเพจมีเนื้อหาตั้งแต่การโหลดครั้งแรก จึงทำให้เว็บเพจสามารถติดอันดับการค้นหาใน Google หรือ Search Engine อื่น ๆ ได้ง่ายขึ้นด้วย

บทวนเกี่ยวกับ React

React คือ JavaScript ไลบรารีที่ถูกพัฒนาขึ้นโดย Facebook มีจุดประสงค์สำหรับสร้างส่วนติดต่อกับผู้ใช้ (User Interface หรือ UI) บนเว็บเพจหรือบนเว็บแอปพลิเคชัน

React จะแบ่งเว็บเพจออกเป็นส่วนย่อย เรียกว่า คอมโพเนนต์ (Components) นักพัฒนาสามารถนำคอมโพเนนต์ต่าง ๆ มาประกอบเป็นหน้าเว็บเพจ หากหน้าเพจมีเนื้อหาซ้ำก็สามารถนำคอมโพเนนต์เก่ามาใช้งานได้ ด้วยการแบ่งเนื้อหาออกเป็นส่วนย่อย จึงทำให้การพัฒนาแอปพลิเคชันทำได้ง่ายกว่าการเขียนโค้ด JavaScript ด้วยตนเองทั้งหมด

React จะใช้วิธีเก็บโครงสร้างของเว็บเพจไว้ในหน่วยความจำ เรียกว่า Virtual DOM โดยจะนำไปเปรียบเทียบกับเอลิเมนต์ที่อยู่ใน DOM จริง ด้วยวิธีนี้ React จึงสามารถอัปเดตเฉพาะส่วนของเว็บเพจที่มีการเปลี่ยนแปลงไปจากเดิม โดยไม่จำเป็นต้องอัปเดตข้อมูลทั้งหน้าเว็บเพจ ทำให้การแสดงผลและการโต้ตอบระหว่างผู้ใช้ในระหว่างใช้งานแอปพลิเคชันสามารถทำได้อย่างรวดเร็วยิ่งขึ้น



ด้วยคุณสมบัติต่าง ๆ ของ React นักพัฒนาจึงนิยมนำ React มาใช้สร้างเว็บแอปพลิเคชันที่ใช้เพียงหน้าเว็บเพจเดียว (เรียกว่า Single Page Application : SPA) ที่สามารถแสดงผลและโต้ตอบกับผู้ใช้โดยไม่ต้องโหลดหน้าเว็บใหม่ ทำให้แอปพลิเคชันทำงานได้รวดเร็วและมีประสิทธิภาพ

จาก React มาสู่ Next.js

React เป็น JavaScript ไลบรารี ที่ใช้งานง่ายและมีประสิทธิภาพสูง จึงเป็นที่นิยมและถูกนำไปใช้งานอย่างกว้างขวาง แต่การนำ React ไปใช้งานจริงยังมีข้อจำกัดบางประการ เช่น ไม่มีการกำหนดแนวทางหรือโครงสร้างใด ๆ มาให้ นักพัฒนาจะต้องออกแบบเอง ดังนั้น หากนักพัฒนาออกแบบโครงสร้างของแอปพลิเคชันไม่ดีพอตั้งแต่แรก ก็จะทำให้การแก้ไขหรือการขยายส่วนประกอบต่าง ๆ ของแอปพลิเคชันทำได้ยาก

React ใช้วิธีเรนเดอร์เนื้อหาในฝั่งไคลเอนต์เป็นหลัก ทำให้เว็บเพจไม่มีข้อมูลที่อธิบายว่าเว็บเพจปัจจุบันมีเนื้อหาเกี่ยวกับอะไร จึงมีปัญหากับการทำ SEO ยากต่อการทำให้เว็บไซต์ติดอันดับการค้นหาใน Google หรือใน Search Engine ต่าง ๆ

เพื่อแก้ปัญหานี้ React เฟรมเวิร์คที่ชื่อว่า Next.js จึงถูกสร้างขึ้น โดยนำ React มาปรับปรุงใหม่ เพื่อให้สามารถสร้างแอปพลิเคชันสำหรับการใช้งานในโลกแห่งความจริงได้ดียิ่งขึ้น

Next.js ถูกพัฒนาขึ้นโดย Vercel และถูกเผยแพร่ครั้งแรกในปี ค.ศ. 2017 ซึ่ง Next.js ยังคงใช้รูปแบบพื้นฐานของ React เอาไว้ เช่น ใช้ JSX สำหรับสร้างเนื้อหาของคอมโพเนนต์, ส่งผ่านข้อมูลระหว่างคอมโพเนนต์โดยใช้ props, การจัดการ state และการใช้ฮุค ฯลฯ เรียกได้ว่านักพัฒนาที่คุ้นเคยกับ React มาก่อนแล้ว ก็สามารถเรียนรู้การใช้งาน Next.js ได้อย่างรวดเร็ว

จากที่กล่าวมาสามารถสรุปได้ว่า Next.js ก็คือ เครื่องมือสำหรับการพัฒนาเว็บแอปพลิเคชันด้วย React ในแบบมืออาชีพ ที่สามารถนำแอปพลิเคชันไปใช้งานจริงได้นั่นเอง

คุณสมบัติพื้นฐานของ Next.js

Next.js คือ JavaScript เฟรมเวิร์ค ซึ่งพัฒนาต่อยอดมาจาก React โดยมีคุณสมบัติหลัก ๆ ดังนี้

- Typescript Support คือ รองรับการเขียนโค้ดด้วย TypeScript โดยไม่ต้องตั้งค่าส่วนประกอบใด ๆ
- Server-Side Rendering (SSR) คือ การสร้างเนื้อหาเว็บเพจในฝั่งเซิร์ฟเวอร์ จากนั้นจึงส่ง HTML ที่สมบูรณ์แล้วไปแสดงผลยังไคลเอนต์ ด้วยวิธีนี้การโหลดหน้าเว็บเพจในครั้งแรกจึงทำได้เร็ว และยังช่วยในเรื่องของการทำ SEO เพิ่มอันดับการค้นหาใน Google หรือ Search Engine อื่น ๆ
- Static Site Generation (SSG) คือ การสร้างหน้าเว็บเตรียมเอาไว้ล่วงหน้าตั้งแต่ตอนสร้างแอปพลิเคชัน (Build Time) เมื่อผู้ใช้ต้องการดูหน้าเว็บเพจ เซิร์ฟเวอร์สามารถส่งผลลัพธ์ไปแสดงผลได้ทันทีโดยไม่ต้องรอประมวลผล
- Code Splitting คือ การแบ่งโค้ดออกเป็นส่วนเล็ก ๆ (เรียกว่า chunks) เพื่อลดขนาดของไฟล์ ทำให้เว็บไซต์โหลดเร็วขึ้นและประหยัดทรัพยากร
- Image Optimization คือ ระบบจัดการกับรูปภาพ ที่สามารถปรับรูปภาพให้มีขนาดและความละเอียดที่เหมาะสมตามสภาพแวดล้อมในขณะนั้น ๆ

- File-base Routing คือ การกำหนดระบบเราท์ติ้งโดยใช้ชื่อโฟลเดอร์และโครงสร้างการจัดเรียงโฟลเดอร์ ซึ่งวิธีนี้ช่วยให้การจัดโครงสร้างของแอปพลิเคชันสามารถทำได้ง่าย
- API Routes คือ รองรับการสร้าง API endpoints ได้ภายในโปรเจกต์ ทำให้สามารถสร้างเว็บแอปพลิเคชันทั้งในแบบ Frontend และ Backend โดยไม่ต้องติดตั้งส่วนประกอบเพิ่มเติม
- Hot Module Replacement (HMR) คือ คุณสมบัติในการแสดงผลลัพธ์หลังเปลี่ยนแปลงโค้ดในทันที โดยไม่ต้องรีเฟรชหน้าเว็บเพจ
- Built-In CSS and Sass Support คือ สนับสนุนการปรับแต่งสไตล์ด้วย CSS และ Sass รวมถึงสนับสนุนการใช้งาน CSS-in-JS ร่วมกับไลบรารีต่าง ๆ

เตรียมพร้อมก่อนใช้งาน Next.js

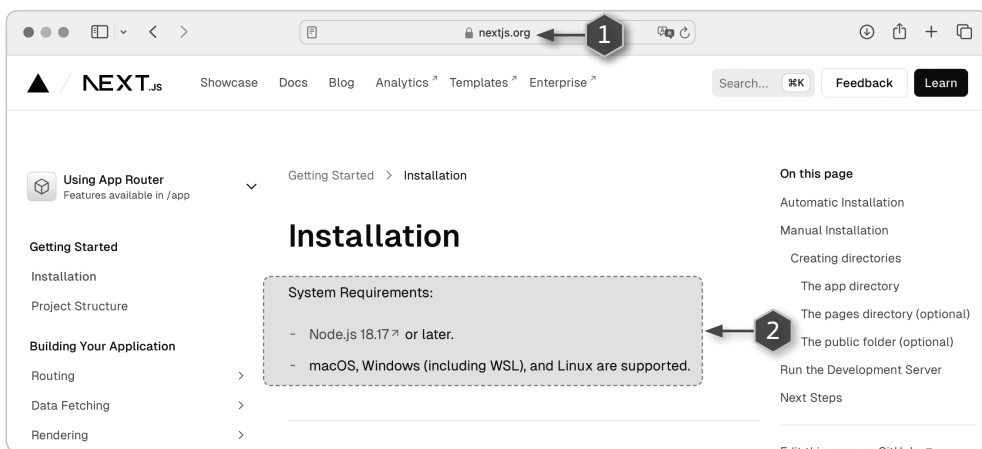
ก่อนใช้งาน Next.js แนะนำให้อ่านเอกสารเกี่ยวกับคุณสมบัติพื้นฐานของแต่ละเวอร์ชัน และติดตั้งซอฟต์แวร์หรือเครื่องมือบางอย่างลงในเครื่องคอมพิวเตอร์ ดังนี้

- ขั้นตอนที่ 1 อ่านเอกสารจาก nextjs.org
- ขั้นตอนที่ 2 ติดตั้ง Node.js
- ขั้นตอนที่ 3 ติดตั้ง IDE หรือ Code Editor สำหรับใช้แก้ไขโค้ดต่าง ๆ

ขั้นตอนที่ 1 อ่านเอกสารจาก nextjs.org

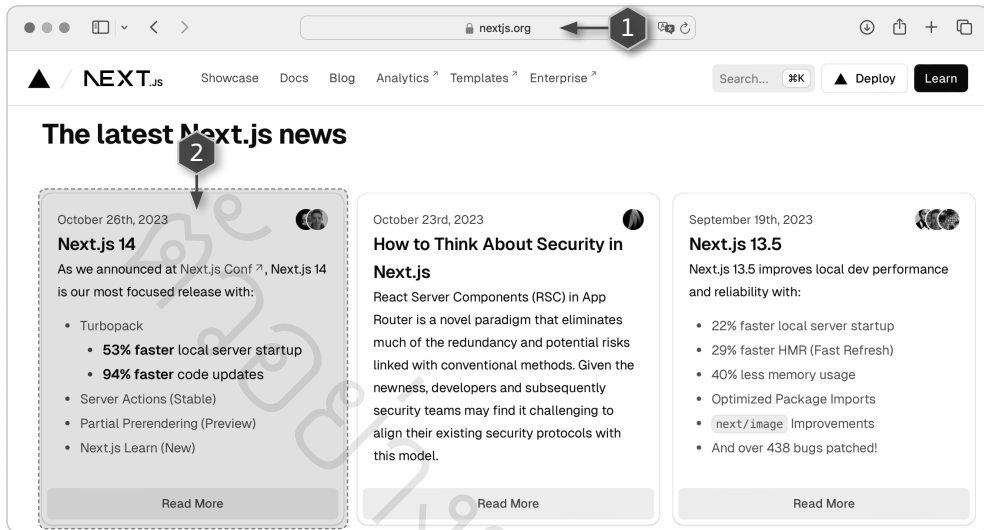
เนื่องจาก Next.js ถูกพัฒนาตลอดเวลา ดังนั้น ก่อนใช้งาน Next.js ทุกครั้งจะต้องตรวจสอบเวอร์ชัน และความต้องการของระบบที่ใช้กับเวอร์ชันนั้น ๆ เสียก่อน

1. ไปที่เว็บ <https://nextjs.org/docs/getting-started/installation>
2. ให้สังเกตในส่วนของ System Requirements ซึ่งบอกถึงความต้องการของระบบที่จะต้องใช้กับ Next.js มีอะไรบ้าง ซึ่งในตัวอย่าง การใช้ Next.js 14 จะต้องติดตั้ง Node.js เวอร์ชัน 18.17 ขึ้นไป โดยรองรับกับระบบปฏิบัติการแมคโอเอส, วินโดวส์ (ที่สนับสนุน WSL) และระบบปฏิบัติการลินุกซ์

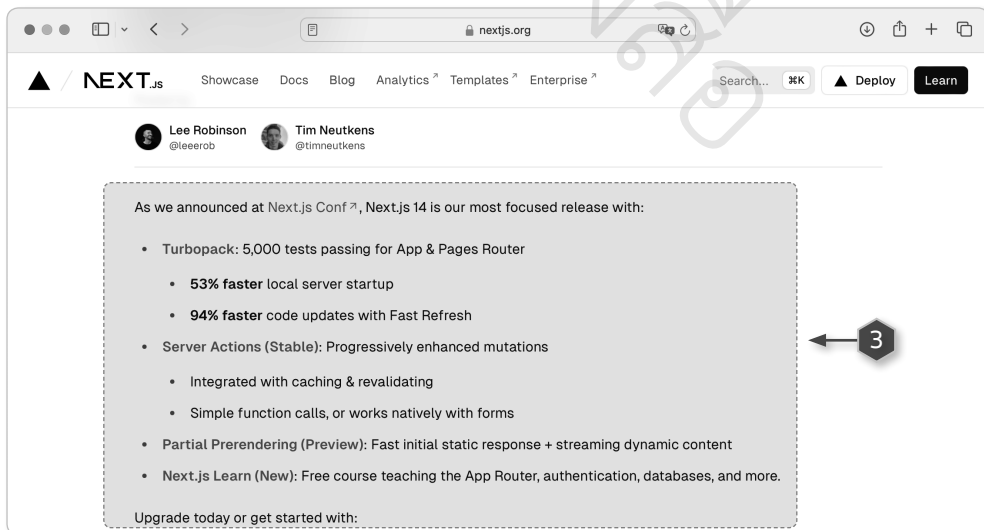


นอกจากต้องทราบความต้องการพื้นฐานแล้ว นักพัฒนาจะต้องทราบว่า Next.js เวอร์ชันที่ต้องการใช้งานนั้นมีคุณสมบัติอะไรบ้างที่ถูกประกาศใช้งานอย่างเป็นทางการ (stable) และคุณสมบัติใดบ้างที่อยู่ในขั้นตอนการพัฒนา (preview, alpha หรือ beta)

1. เปิดเบราว์เซอร์และไปยัง <https://nextjs.org/blog> เพื่อดูข่าวสารที่ทาง Next.js ประกาศออกมา
2. คลิกดูรายละเอียดในข้อมูล หรือเวอร์ชันที่สนใจ เช่น Next.js 14



3. จะแสดงรายละเอียดเกี่ยวกับเวอร์ชันแสดงให้เห็นทราบ เช่น ใน Next.js 14 สามารถใช้คุณสมบัติ Server Actions ได้ เนื่องจากผ่านการทดสอบว่าเสถียรแล้ว (Stable) ส่วนคุณสมบัติ Partial Prerendering ยังอยู่ในขั้นตอนการทดสอบในเบื้องต้น (Preview) จึงอาจพบปัญหาในระหว่างใช้งาน

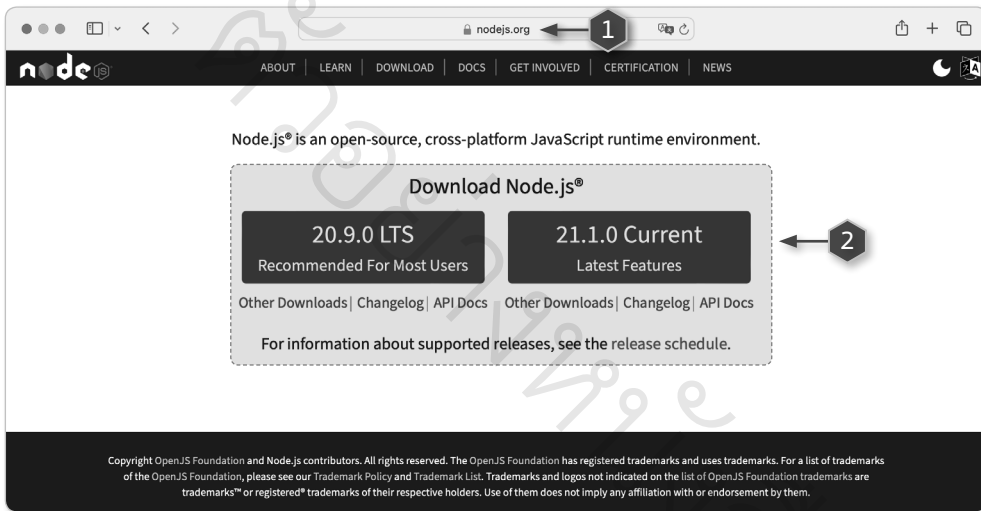


ขั้นตอนที่ 2 ติดตั้ง Node.js

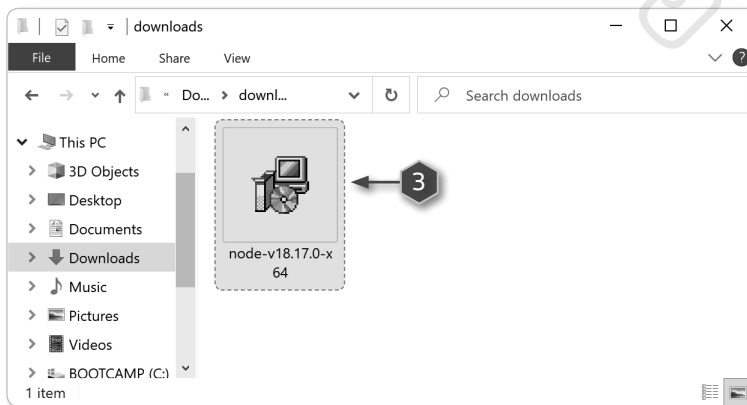
Node.js เป็นแพลตฟอร์มที่อนุญาตให้นักพัฒนาเขียนโค้ด JavaScript (หรือ TypeScript) สำหรับติดต่อกับระบบปฏิบัติการ หรือสร้างแอปพลิเคชันในฝั่งเซิร์ฟเวอร์ได้

เนื่องจาก Next.js ใช้คุณสมบัติพื้นฐานบางอย่างมาจาก Node.js ดังนั้น เราจึงต้องติดตั้ง Node.js ลงไปในเครื่องคอมพิวเตอร์เสียก่อน

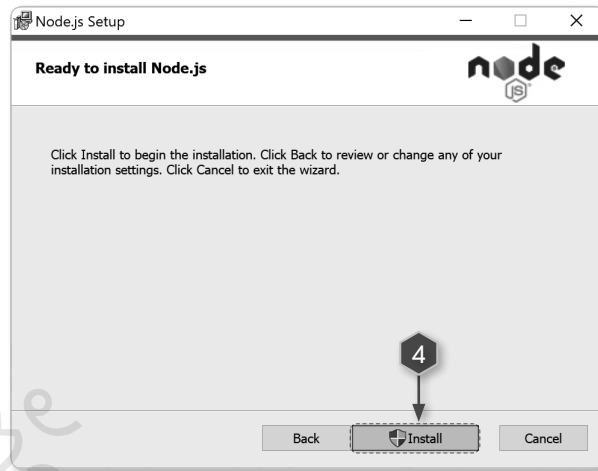
1. ไปที่เว็บไซต์ <https://nodejs.org/>
2. คลิกเพื่อดาวน์โหลด Node.js โดยมี 2 เวอร์ชันให้เลือก ได้แก่ เวอร์ชันที่เป็น LTS (Long Term Support) ซึ่งหมายถึงเวอร์ชันที่ถูกใช้งานอย่างแพร่หลายโดยไม่พบปัญหา กับเวอร์ชัน Current ที่หมายถึงเวอร์ชันล่าสุดในปัจจุบัน โดยส่วนตัวของผู้เขียนแนะนำให้ดาวน์โหลดเวอร์ชันที่เป็น LTS



3. หลังจากดาวน์โหลดเรียบร้อยแล้ว ให้ดับเบิลคลิกเพื่อติดตั้ง Node.js



4. ทำตามขั้นตอนที่ปรากฏบนหน้าจอจนกระทั่งติดตั้งเรียบร้อยแล้ว

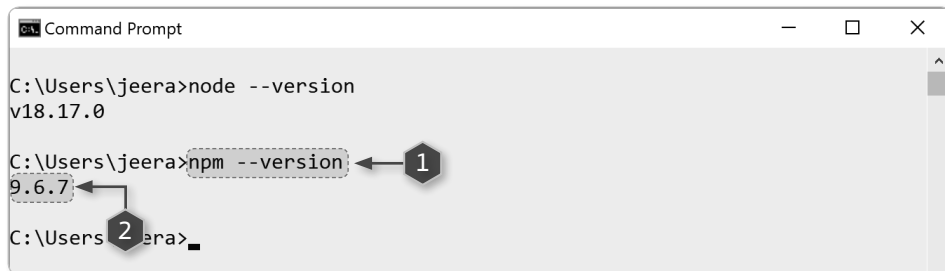


5. เปิดหน้าต่าง Command Prompt ของวินโดวส์ หรือหน้าต่าง Terminal ในแมคโอเอส จากนั้นพิมพ์คำสั่ง `node --version` หรือ `node -v` เพื่อตรวจสอบเวอร์ชันของ Node.js ถ้าการติดตั้ง Node.js ถูกต้องสมบูรณ์จะพบหมายเลขเวอร์ชันแสดงให้ทราบ



หลังจากติดตั้ง Node.js เราจะได้เครื่องมือบางอย่างจาก Node.js มาใช้งาน เช่น ได้เครื่องมือ Node Package Manager หรือ npm สำหรับจัดการกับแพ็คเกจต่าง ๆ เช่น ใช้ติดตั้งแพ็คเกจ, ถอดแพ็คเกจ หรืออัปเดตแพ็คเกจ หากต้องการตรวจสอบว่า npm ได้ถูกติดตั้งเรียบร้อยแล้ว สามารถตรวจสอบจากหมายเลขเวอร์ชันของ npm ได้ ดังนี้

1. เปิดหน้าต่าง Command Prompt ของวินโดวส์ หรือหน้าต่าง Terminal ในแมคโอเอส จากนั้นพิมพ์คำสั่ง `npm --version` หรือ `npm -v` เพื่อตรวจสอบเวอร์ชันของ npm
2. ถ้าการติดตั้ง npm ถูกต้องสมบูรณ์จะพบหมายเลขเวอร์ชันของ npm แสดงให้ทราบ



NOTE\\>

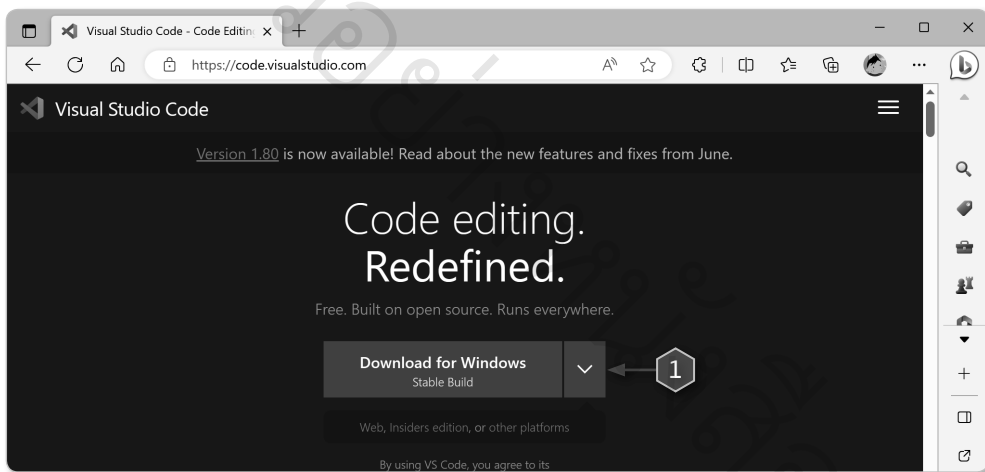
หากพบปัญหาเกี่ยวกับการใช้งาน npm ให้ติดตั้ง npm เข้าไปใหม่โดยใช้คำสั่ง `npm install npm -g`

ขั้นตอนที่ 3 ติดตั้ง Visual Studio Code

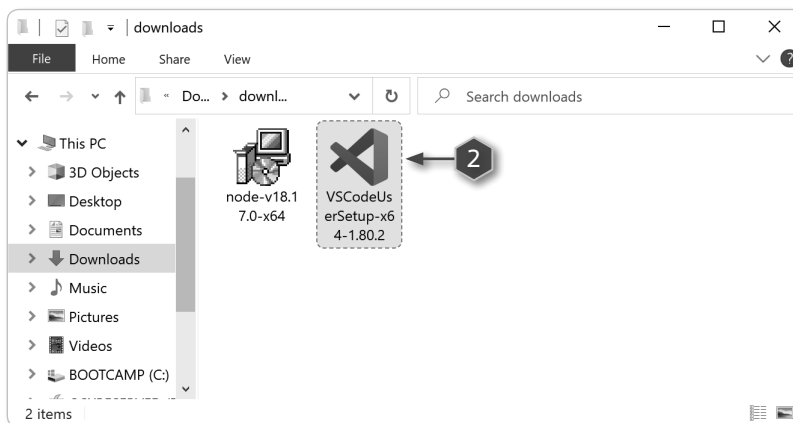
Visual Studio Code (หรือเรียกสั้น ๆ ว่า VS Code) เป็น Code Editor ยอดนิยมาจากทางไมโครซอฟท์ ที่มีผู้ใช้งานเป็นจำนวนมากทั่วโลก โดยมีจุดเด่นมากมาย เช่น สามารถดาวน์โหลดไปใช้งานได้ฟรี, สามารถติดตั้งส่วนประกอบเสริมจึงเพิ่มความสามารถได้ไม่จำกัด, มีตัวช่วยกรอกคำสั่ง, มีการใช้สีเพื่อแยกโค้ดคำสั่งแต่ละแบบจึงทำให้แก้ไขโค้ดหรืออ่านโค้ดได้ง่าย และที่สำคัญคือ สามารถใช้กับภาษาคอมพิวเตอร์ได้เกือบทุกภาษา ไม่ว่าจะเป็น C, Java, PHP, Python, JavaScript และภาษาอื่น ๆ อีกมากมาย

การติดตั้ง VS Code เพื่อใช้งานกับ JavaScript (หรือ TypeScript) มีขั้นตอน ดังนี้

1. ไปที่เว็บไซต์ <https://code.visualstudio.com> คลิกปุ่ม Download



2. ดับเบิลคลิกไฟล์ติดตั้ง และทำตามขั้นตอนที่ปรากฏบนหน้าจอ



3. คลิกปุ่ม Finish เป็นขั้นตอนสุดท้าย



เพื่อความสะดวกในการพัฒนา Next.js ร่วมกับ Visual Studio Code แนะนำให้ติดตั้งส่วนประกอบเสริม (extensions หรือ plug-in) ต่อไปนี้ลงไป

- JavaScript (ES6) code snippets เป็นส่วนประกอบเสริมที่จะช่วยให้เราสามารถเขียนโค้ด JavaScript สมัยใหม่ (ES6 ขึ้นไป) ทำได้ง่ายและรวดเร็ว โดยในระหว่างที่กรอกโค้ดจะมีคำแนะนำเกี่ยวกับชุดคำสั่งหรือแสดงโครงสร้างโค้ดพื้นฐานมาให้อัตโนมัติ
- Prettier - Code formatter เป็นเครื่องมือสำหรับการจัดรูปแบบโค้ดโดยอัตโนมัติ เช่น มีการปรับการเยื้อง ทำให้โค้ดมีระเบียบและอ่านง่าย
- Tailwind CSS IntelliSense เป็นเครื่องมือที่ช่วยแนะนำชื่อย่อ (utilities class) ของ Tailwind CSS มาให้อัตโนมัติในระหว่างกรอกโค้ด พร้อมแสดงคำอธิบายโค้ด CSS และสามารถซ่อนชื่อย่ออัตโนมัติ (หนังสือเล่มนี้ใช้วิธีปรับแต่งหน้าตาของแอปพลิเคชันด้วย Tailwind CSS ดังนั้น จึงแนะนำให้ติดตั้งส่วนประกอบเสริมนี้ด้วย)

การติดตั้งส่วนประกอบเสริม (Extensions) เพิ่มเข้าไปใน VS Code มีขั้นตอน ดังนี้

1. เปิดโปรแกรม VS Code จากนั้นให้คลิกปุ่ม  หรือกดปุ่ม <Ctrl+Shift+X> (ในแมคให้กดปุ่ม <Cmd+Shift+X>) เพื่อแสดงส่วนประกอบเสริมทั้งหมดที่ได้ติดตั้งเอาไว้
2. ที่ช่องค้นหา ให้กรอก JavaScript เพื่อค้นหาส่วนประกอบเสริมสำหรับ JavaScript
3. จะพบส่วนประกอบเสริมมากมายที่เกี่ยวกับ JavaScript คลิกปุ่ม Install ที่ JavaScript (ES6) code snippets เพื่อติดตั้งส่วนประกอบต่าง ๆ ที่จำเป็นต่อการใช้ Visual Studio Code ร่วมกับ JavaScript

Part 2

Essential
Next.js

08 | การใช้งานเราท์ติ้งใน Next.js

Routing (เราท์ติ้ง) คือ การกำหนดว่าเมื่อผู้ใช้เปลี่ยนมายังเส้นทาง (route : เราท์) ที่กำหนด จะให้แอปพลิเคชันตอบสนองต่อการเปลี่ยนแปลงเราท์นั้นอย่างไร เช่น เมื่อผู้ใช้มายังเราท์ '/products' ก็ให้แสดงข้อมูลสินค้าทั้งหมด แต่ถ้าผู้ใช้มายังเราท์ '/products/1' ก็ให้แสดงสินค้าที่มีรหัสสินค้าเท่ากับ 1 เป็นต้น

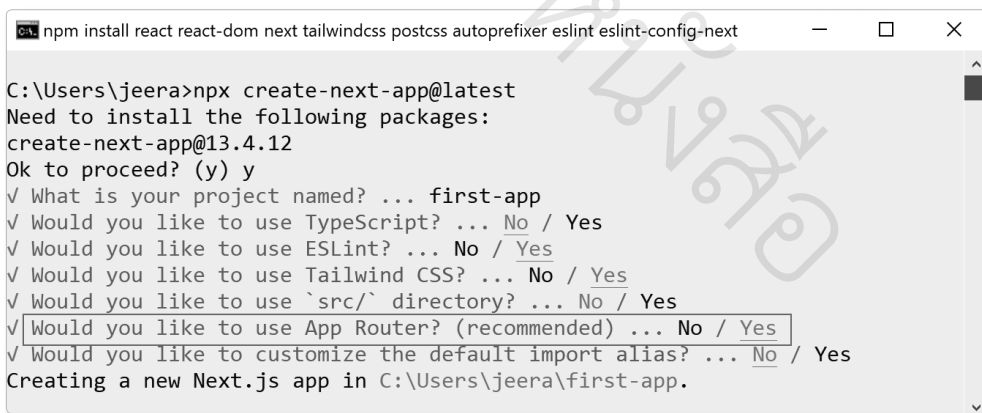
ใน Next.js จะกำหนดเราท์ด้วยวิธี File-base Routing ซึ่งหมายถึง การกำหนดเราท์โดยดูจากการจัดเรียงโฟลเดอร์และไฟล์ภายในโปรเจกต์ ด้วยวิธีนี้ช่วยให้นักพัฒนาไม่ต้องกังวลเกี่ยวกับการตั้งค่าหรือตรวจสอบเส้นทางต่าง ๆ เพราะเพียงแค่กำหนดโฟลเดอร์ และกำหนดไฟล์ตามกฎหมายของ Next.js ก็จะได้ระบบเราท์ที่ใช้งานได้ใช้งานโดยอัตโนมัติ

App Router และ Pages Router

ใน Next.js เวอร์ชัน 13.4 ขึ้นไป สามารถเลือกระบบเราท์ได้ 2 แบบ ได้แก่ App Router และ Pages Router

โดย Pages Router ถือว่าเป็นคุณสมบัติพื้นฐานของ Next.js ที่มาพร้อมกับ Next.js เวอร์ชันแรก ส่วน App Router เป็นคุณสมบัติใหม่ที่เพิ่งประกาศใช้งานใน Next.js 13.4

ในขั้นตอนการสร้างโปรเจกต์ใหม่ โดยใช้คำสั่ง `npx create-next-app@latest` นักพัฒนาสามารถเลือกติดตั้ง App Router หรือไม่ก็ได้ ดังรูป



```

C:\Users\jeera>npx create-next-app@latest
Need to install the following packages:
create-next-app@13.4.12
Ok to proceed? (y) y
√ What is your project named? ... first-app
√ Would you like to use TypeScript? ... No / Yes
√ Would you like to use ESLint? ... No / Yes
√ Would you like to use Tailwind CSS? ... No / Yes
√ Would you like to use `src/` directory? ... No / Yes
√ Would you like to use App Router? (recommended) ... No / Yes
√ Would you like to customize the default import alias? ... No / Yes
Creating a new Next.js app in C:\Users\jeera\first-app.
  
```

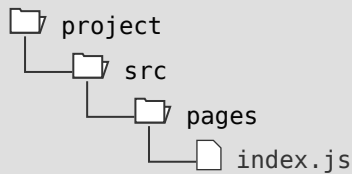
▲ รูปแสดงการเลือกติดตั้ง App Router

กรณีที่นักพัฒนาเลือกใช้งาน App Router ระบบเราท์ติ้งจะถูกกำหนดไว้ในโฟลเดอร์ `app` และหน้าโฮมเพจจะถูกกำหนดไว้ที่ไฟล์ `/app/page.js`





แต่ถ้าตอนสร้างโปรเจกต์ใหม่ไม่ได้เลือกใช้งาน App Router ระบบเร้าต์ติ้งจะถูกกำหนดไว้ที่โฟลเดอร์ pages และหน้าโฮมเพจจะถูกกำหนดไว้ที่ไฟล์ /pages/index.js



ในหนังสือเล่มนี้จะใช้ App Router เป็นหลัก เนื่องจากเป็นระบบเร้าต์ติ้งที่พัฒนาขึ้นใหม่ สำหรับใช้งานร่วมกับ Server Component และ Client Component ซึ่งเป็นคุณสมบัติใหม่ของ React และ Next.js

NOTE\\>

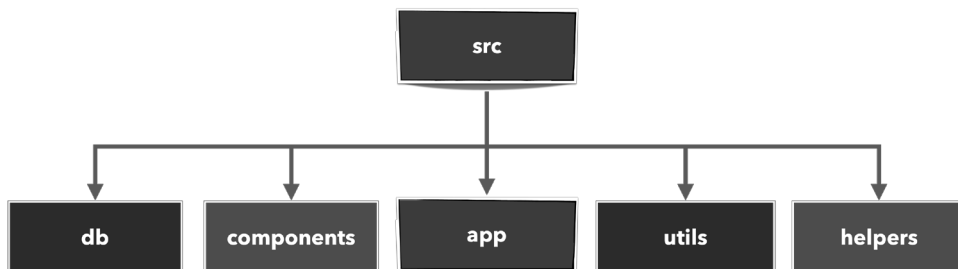
ในขั้นตอนการสร้างโปรเจกต์ใหม่ นักพัฒนาสามารถเลือกที่จะเก็บแอปพลิเคชันไว้ในโฟลเดอร์ src หรือไม่ได้ กรณีที่เราไม่ได้เลือกใช้ โฟลเดอร์ app จะถูกกำหนดให้อยู่ภายใต้โฟลเดอร์ที่เป็นชื่อโปรเจกต์ ดังรูป



โครงสร้างของ App Router ใน Next.js

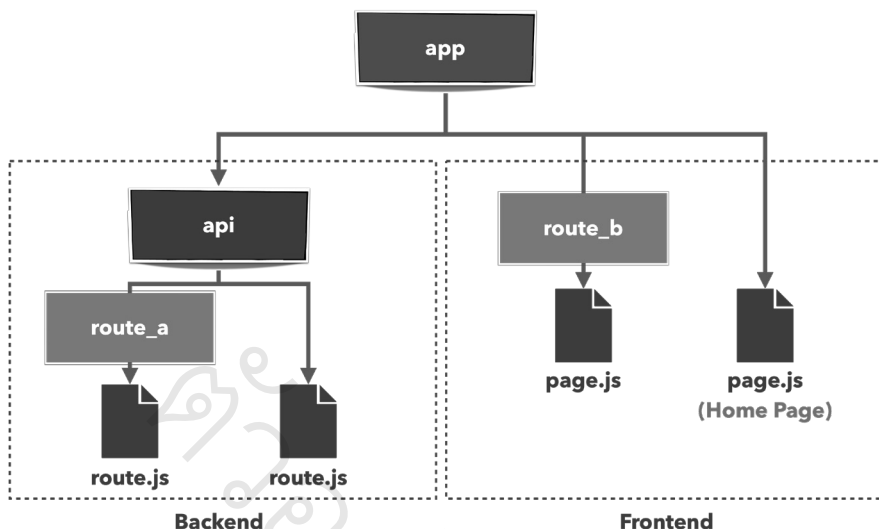
ใน Next.js จะกำหนดโฟลเดอร์ src สำหรับเก็บข้อมูลแอปพลิเคชันทั้งหมด และกำหนดโฟลเดอร์ app สำหรับเก็บการทำงานของแอปพลิเคชัน

ภายในโฟลเดอร์ src นักพัฒนาสามารถกำหนดโฟลเดอร์อื่น ๆ ขึ้นมาเอง เช่น โฟลเดอร์ components, โฟลเดอร์ db, โฟลเดอร์ utils หรือโฟลเดอร์ helpers ฯลฯ



การกำหนดเร้าต์ติ้งใน Next.js จะใช้วิธีที่เรียกว่า File-based Routing ซึ่งหมายถึง การกำหนดเร้าต์ติ้งโดยใช้ลำดับการจัดเรียงไฟล์และโฟลเดอร์ ซึ่งหนังสือเล่มนี้เลือกใช้ App Router ดังนั้น การกำหนดเร้าต์ติ้งจะอยู่ภายใต้โฟลเดอร์ app ทั้งหมด

ในโพลีเตอร์ app สามารถแบ่งแอปพลิเคชันเป็น 2 ส่วน ได้แก่ Frontend และ Backend ซึ่ง Next.js จะกำหนดให้ส่วนของ Backend ต้องอยู่ภายใต้โพลีเตอร์ api ดังรูป



Next.js ได้กำหนดกฎสำหรับการใช้งาน Frontend และ Backend ดังนี้

- กำหนดไฟล์ page.js สำหรับใช้แสดงหน้าเว็บเพจ ใช้เพื่อสร้างส่วนติดต่อกับผู้ใช้ (UI) ที่อยู่ในฝั่ง Frontend เช่น เมนู รูปภาพ ข้อมูลสินค้า แผนที่ และแบบฟอร์ม เป็นต้น
- กำหนดไฟล์ route.js สำหรับจัดการกับ Request ที่ส่งมาจากไคลเอนต์ (Request Handler) เช่น ถ้าผู้ใช้ส่งการร้องขอข้อมูลมายังเราท์ '/api/products' ภายในไฟล์ route.js ก็จะทำหน้าที่ติดต่อกับฐานข้อมูลและส่งข้อมูลสินค้ากลับไปที่ไคลเอนต์

พื้นฐานการใช้งาน App Router

App Router เป็นคุณสมบัติใหม่ที่ถูกประกาศใช้งานใน Next.js 13.4 ใช้กำหนดระบบเราท์ติ้งให้กับแอปพลิเคชัน โดยมีคุณสมบัติที่น่าสนใจหลายประการ เช่น

- สามารถกำหนดเราท์ติ้งที่ซับซ้อนได้ (nested routing)
- สามารถกำหนดเค้าโครงให้กับหน้าเว็บเพจ ได้แก่ การใช้ layout.js และ template.js
- สามารถแสดงสถานะโหลดข้อมูล (loading states) เช่น การกำหนด <Suspend></Suspend>
- มีระบบตรวจสอบความผิดพลาด (error-handling)
- การ fetch ข้อมูลจากฐานข้อมูล หรือจาก API สามารถเก็บข้อมูลลงแคช ทำให้การโหลดข้อมูลในครั้งต่อไปทำได้เร็วขึ้น
- สามารถเลือกเรนเดอร์เนื้อหาแสดงผลจากฝั่งไคลเอนต์ หรือฝั่งเซิร์ฟเวอร์ก็ได้
- สามารถเตรียมเนื้อหา (Pre-rendering) ตั้งแต่ตอนสร้างแอปพลิเคชัน (Build Time) หรือเรนเดอร์เนื้อหาในแบบเรียลไทม์
- และอื่น ๆ

URL, URL Path และ URL Segment

คำศัพท์ที่จำเป็นต้องทราบก่อนการใช้งานระบบเร้าต์ติ้งของ Next.js ได้แก่ URL, URL Path และ URL Segment

URL (Uniform Resource Locator) คือ ที่อยู่ (แอดเดรส) ที่ใช้สำหรับเข้าถึงหน้าเว็บ หรือเข้าถึงทรัพยากรอื่น ๆ บนอินเทอร์เน็ต ซึ่งประกอบไปด้วย 3 ส่วนหลัก ๆ ได้แก่ โพรโทคอล (http หรือ https), ชื่อโดเมน (เช่น www.thinkbeyondbook.com) และ Path

https://www.thinkbeyondbook.com/site/catalog

Protocol

Domain Name

Path

URL Path หรือมักเรียกสั้น ๆ ว่า Path (พาส) คือ ส่วนหนึ่งของ URL ที่อยู่ด้านหลังชื่อโดเมน โดยพาสจะเป็นเส้นทาง (Route) ซึ่งบอกให้ทราบถึงตำแหน่งของเว็บเพจปัจจุบัน ว่าอยู่ส่วนไหนในโครงสร้างของเว็บไซต์ จากตัวอย่าง ที่แอดเดรส ‘https://www.thinkbeyondbook.com/site/book/1933’ ส่วนของ ‘/site/book/1933’ คือ พาสที่ใช้อ้างอิงตำแหน่งของเว็บเพจปัจจุบัน

URL Segment คือ ส่วนหนึ่งของ URL Path ที่ถูกคั่นด้วยเครื่องหมาย ‘/’ หรืออาจกล่าวอย่างง่าย ๆ ว่า URL Segment ก็คือส่วนย่อยที่ประกอบขึ้นเป็น Path นั่นเอง

ดังนั้น แอดเดรส ‘https://www.thinkbeyondbook.com/site/book/1933’ ส่วนที่เป็น URL Path คือ ‘/site/book/1933’ จะประกอบด้วย URL Segment จำนวน 3 ส่วน ได้แก่ ‘site’, ‘book’ และ ‘1933’

วิธีกำหนดเร้าต์ใน App Router

Route (เร้าต์) ในความหมายของ Next.js คือ การใช้พาส (URL Path) เพื่อกำหนดเนื้อหาที่จะนำมาแสดงยังเว็บเพจ โดยปกติเมื่อพาสต่างกัน ข้อมูลหรือเนื้อหาที่แสดงบนเว็บเพจก็จะแตกต่างกัน ยกเว้นการกำหนดพาสในแบบไดนามิก ที่แม้ว่าจะใช้พาสเดียวกันแต่เนื้อหาก็สามารถแตกต่างกันได้ ขึ้นอยู่กับค่าพารามิเตอร์ที่ส่งมากับพาส

การกำหนดเร้าต์ใน App Router จะใช้วิธีที่เรียกว่า File-based Routing ซึ่งหมายถึง การกำหนดเร้าต์โดยใช้ลำดับการจัดเรียงไฟล์และโฟลเดอร์ที่อยู่ภายในโฟลเดอร์ app

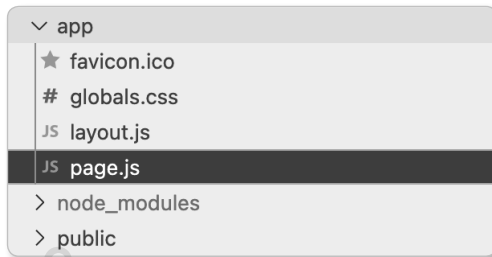
การกำหนดเร้าต์ใน Next.js จะต้องทำตามเงื่อนไข ดังต่อไปนี้

1. ชื่อโฟลเดอร์ที่กำหนดภายในโฟลเดอร์ app จะถูกนำไปใช้เป็นชื่อเร้าต์ เมื่อมีการกำหนดโฟลเดอร์ย่อยชื่อโฟลเดอร์จะถูกนำไปใช้เป็นชื่อเร้าต์ย่อยด้วย
2. ไฟล์ page.js (page.jsx, page.ts หรือ page.tsx) ใช้กำหนดเนื้อหาที่จะถูกนำมาแสดงในแต่ละเร้าต์ ดังนั้น หากในแอปพลิเคชันมีการกำหนดเร้าต์มากกว่า 1 เร้าต์ ก็ต้องกำหนดไฟล์ page.js หลายไฟล์ เพื่อแสดงผลลัพธ์ที่แตกต่างกัน

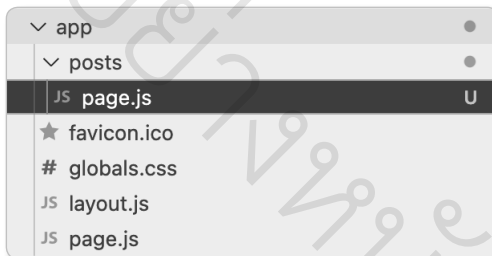
จากหลักเกณฑ์ข้างต้น สรุปได้ว่า ชื่อโฟลเดอร์ภายในโฟลเดอร์ app จะถูกนำไปใช้เป็นชื่อเร้าต์ ส่วนไฟล์ page.js (หรือไฟล์ page.jsx, page.ts หรือ page.tsx) จะถูกใช้เพื่อสร้างเนื้อหา (UI) ให้กับเร้าต์นั้น ๆ

ต่อไปนี้เป็นตัวอย่างการสร้างโฟลเดอร์ และไฟล์ `page.js` เพื่อกำหนดระบบเราท์ติ้ง

- เมื่อสร้างไฟล์ `@/app/page.js` (สร้างโฟลเดอร์ `page.js` ลงในโฟลเดอร์ `app`) Next.js จะกำหนดเราท์เป็น `'/'` ซึ่งหมายถึงเส้นทางที่บอกให้เซิร์ฟเวอร์ใช้แสดงหน้าโฮมเพจ ส่วนเนื้อหาของหน้าโฮมเพจจะถูกกำหนดไว้ในไฟล์ `page.js`



- เมื่อสร้างไฟล์ `/app/posts/page.js` (สร้างโฟลเดอร์ `posts` ไว้ภายในโฟลเดอร์ `app` และกำหนดไฟล์ `page.js` ไว้ภายในโฟลเดอร์ `posts`) ในกรณีนี้ Next.js จะกำหนดเราท์ `'/posts'` มาให้ โดยเนื้อหาของเราท์จะถูกกำหนดไว้ในไฟล์ `page.js`



ตัวอย่าง 8.1

แสดงข้อมูลสินค้า

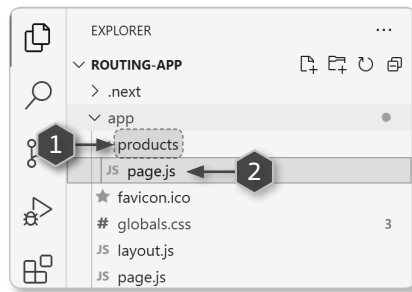
ใน Next.js เวอร์ชัน 13.4 หรือเวอร์ชันที่สูงกว่า เมื่อสร้างโฟลเดอร์ ไว้ภายในโฟลเดอร์ `app` ก็จะถูกกำหนดเป็นชื่อเราท์โดยอัตโนมัติ เช่น สร้างโฟลเดอร์ `products` ลงในโฟลเดอร์ `app` ก็จะกำหนดเราท์ `'/products'` มาให้อัตโนมัติ ส่วนข้อมูลสินค้าที่ต้องการนำมาแสดงผล จะถูกกำหนดไว้ที่ไฟล์ `page.js` ซึ่งอยู่ภายในโฟลเดอร์ `products`

1. เปิดเบราว์เซอร์ไปยังแอดเดรส `http://localhost:3000/products` เพื่อแสดงข้อมูลสินค้า
2. จะพบรายการสินค้าทั้งหมดที่ได้กำหนดไว้ในไฟล์ `page.js`



เพื่อให้ได้ผลลัพธ์ตามตัวอย่าง ให้แก้ไขโปรเจกต์ด้วยขั้นตอน ต่อไปนี้

1. สร้างโฟลเดอร์ products ไว้ภายใต้โฟลเดอร์ app เพื่อกำหนดเร้าก์ '/products'
2. สร้างไฟล์ page.js ไว้ภายใต้โฟลเดอร์ products (ไฟล์ @/app/products/page.js) เมื่อผู้ใช้เปิดเบราว์เซอร์และไปยังแอดเดรส <http://localhost:3000/products> ก็จะแสดงเนื้อหาที่อยู่ในไฟล์ page.js นี้โดยอัตโนมัติ



3. แก้ไขโค้ดในไฟล์ page.js สำหรับแสดงรายการสินค้า (ไฟล์ app/products/page.js)

```
1 import React from 'react'
2 const allProducts = [
3   {id: '1', productName: 'Product A', price: 120},
4   {id: '2', productName: 'Product B', price: 220},
5   {id: '3', productName: 'Product C', price: 390}
6 ]
7
8 export default function ProductsPage() {
9   return (
10     <div className='flex flex-col items-center justify-start mx-auto h-screen '>
11       <h1 className='text-3xl text-gray-500 m-5'>All Products</h1>
12       {allProducts.map((product) => (
13         <div key={product.id} className='flex flex-row justify-evenly mx-auto w-2/5'>
14           <h1>{product.productName}</h1>
15           <p>{product.price}</p>
16         </div>
17       ))}
18     </div>
19   )
20 }
```

4. รันแอปพลิเคชันในโหมด Development โดยใช้คำสั่ง `npm run dev` จากนั้นเปิดเบราว์เซอร์ไปยังแอดเดรส <http://localhost:3000/products> ก็จะพบรายการสินค้าตามต้องการ

Server Component และ Client Component ใน App Router

ใน Next.js คอมโพเนนต์ที่อยู่ภายใต้โฟลเดอร์ `app` จะถูกกำหนดให้มีค่าเริ่มต้นเป็น Server Component ซึ่งหมายถึง เนื้อหาของคอมโพเนนต์จะถูกเรนเดอร์ในฝั่งเซิร์ฟเวอร์ ด้วยวิธีนี้ HTML จะถูกสร้างขึ้นโดยเซิร์ฟเวอร์ และถูกนำไปแสดงผลที่ไคลเอนต์

แต่ถ้าต้องการกำหนดให้คอมโพเนนต์ที่อยู่ภายใต้โฟลเดอร์ `app` มีลักษณะเป็น Client Component ก็สามารถทำได้โดยใส่ไคเร็กทีฟ `'use client'` ที่ด้านบนสุดของโค้ด ดังตัวอย่าง

1. สร้างโฟลเดอร์ `promotions` ไว้ภายในโฟลเดอร์ `app` เพื่อกำหนดเส้นทาง `'/promotions'`
2. สร้างไฟล์ `page.js` ไว้ภายในโฟลเดอร์ `promotions`



3. แก้ไขไฟล์ `page.js` (ไฟล์ `@/app/promotions/page.js`) โดยให้ใส่ไคเร็กทีฟ `'use client'` ที่ด้านบนเพื่อบอกให้ Next.js ทราบว่าต้องการให้คอมโพเนนต์ปัจจุบันเป็น Client Component เมื่อผู้ใช้มายังแอดเดรส `http://localhost:3000/promotions` ก็จะพบหน้าเพจที่ได้เตรียมไว้ตามต้องการ

```
1 'use client'
2 import React from 'react'
3
4 export default function Page() {
5   return (
6     <div className='text-5xl'>Hello Client Component</div>
7   )
8 }
```

- บรรทัดที่ 1 ใส่ไคเร็กทีฟ `'use client'` เพื่อกำหนดให้เป็น Client Component

NOTE\\\>

หากภายในคอมโพเนนต์มีการใช้ฮุค หรือมีการกำหนด Event เพื่อโต้ตอบกับผู้ใช้ คอมโพเนนต์ดังกล่าวจะถูกบังคับให้เป็น Client Component โดยอัตโนมัติ ดังนั้น จึงต้องใส่ `'use client'` กำกับไว้ด้วยเสมอ ไม่งั้นมันจะเกิด error ยกเว้นคอมโพเนนต์ปัจจุบันอยู่ภายใต้ Client Component อื่น ๆ อยู่แล้ว ก็ไม่จำเป็นต้องใส่ `'use client'` กำกับไว้ก็ได้

Part 3

Create Project
with ChatGPT

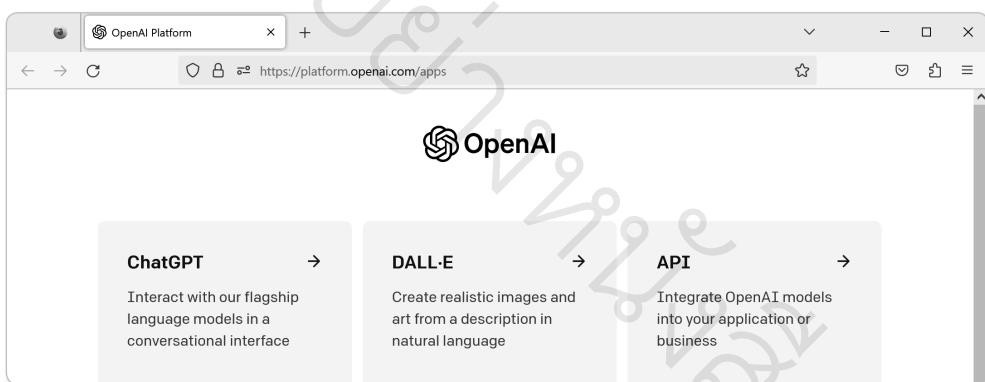
18 | ChatGPT กับการพัฒนาโปรแกรม

หลังจากที่ผู้เขียนได้ทดลองใช้งาน ChatGPT ตั้งแต่เดือนแรกที่ถูกเปิดให้ใช้งานอย่างเป็นทางการ ก็ทราบได้ทันทีว่า ChatGPT จะเป็นเครื่องมือที่ช่วยให้ทุก ๆ คนที่มีพื้นฐานเกี่ยวกับการเขียนโปรแกรม สามารถสร้างโปรแกรมที่ซับซ้อนได้ง่ายขึ้นกว่าเดิมมาก และในอนาคตอันไม่ไกลนี้เราอาจแค่สั่งงานโดยไม่จำเป็นต้องเขียนโปรแกรมเองเลยก็ได้

ในบทนี้ถือว่าการนำประสบการณ์เกี่ยวกับการใช้ ChatGPT ของผู้เขียนมาเล่าสู่กันฟังเท่านั้น ไม่ใช่คำแนะนำ ไม่ใช่รูปแบบที่ต้องทำตาม หรือไม่ใช่อะไรที่ต้องอยู่ในกรอบหรือกฎเกณฑ์ทั้งสิ้น เพราะทุก ๆ คน สามารถสอบถามสิ่งที่อยากทราบจาก ChatGPT ด้วยแนวทางที่ผู้อ่านถนัดได้อยู่แล้วนั่นเอง

บริการต่าง ๆ จาก OpenAI

OpenAI เป็นองค์กรที่ก่อตั้งขึ้นเพื่อการวิจัยและพัฒนาด้านปัญญาประดิษฐ์ (AI) ซึ่งในปัจจุบัน OpenAI ได้เปิดให้บริการด้าน AI อยู่ 3 แบบ ได้แก่ ChatGPT, Dall-E และ API โดยแต่ละแบบมีความแตกต่างกัน ดังนี้

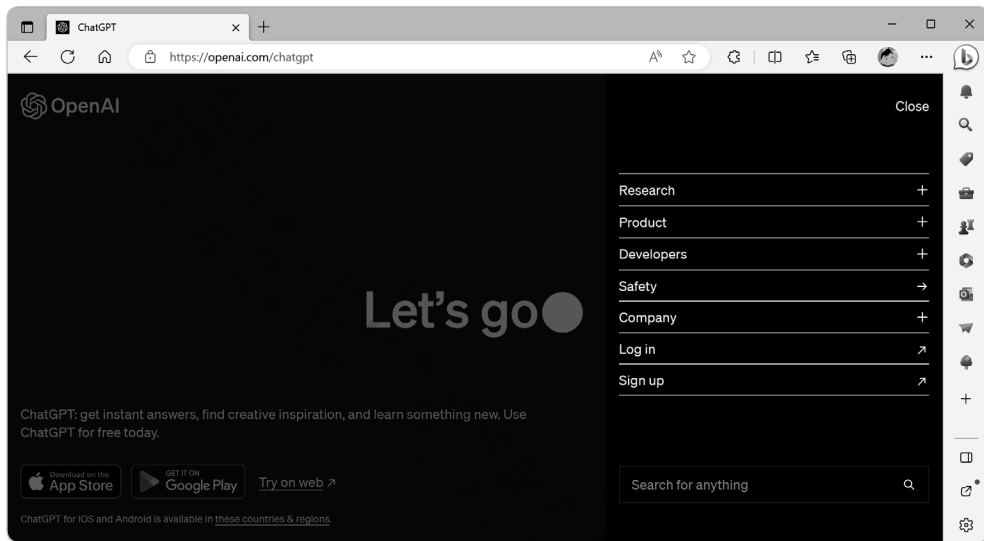


- ChatGPT คือ ปัญญาประดิษฐ์ (AI) ที่ถูกออกแบบมาเพื่อการสนทนา (Chat) โดยในปัจจุบันเปิดบริการเฉพาะการสนทนาด้วยข้อความ ส่วนการสนทนาด้วยเสียงคาดว่าจะเปิดให้บริการอย่างเป็นทางการในอนาคตอันใกล้
- Dall-E คือ ปัญญาประดิษฐ์ที่ถูกออกแบบมาเพื่อสร้างรูปภาพจากคำอธิบายที่เป็นข้อความ ตัวอย่างเช่น ถ้าเราบอกว่า “A highly detailed and sharp image of a car in motion with the Wat Arun temple and the Chao Phraya River in the background.” เป็นการบอกให้ Dall-E สร้างภาพรถยนต์ที่กำลังวิ่ง โดยมีพื้นหลังเป็นวัดอรุณและแม่น้ำเจ้าพระยาอยู่ด้านหลัง และต้องการให้ภาพมีความคมชัดมาก ๆ
- API (OpenAI API) คือ บริการที่ช่วยให้นักพัฒนาสามารถเชื่อมต่อและใช้งาน AI โมเดลต่าง ๆ ของ OpenAI ผ่านทางแอปพลิเคชันจากภายนอกได้ เช่น หากต้องการให้แอปพลิเคชันสามารถ Chat พูดคุยกับลูกค้าโดยอัตโนมัติ ก็สามารถใช้บริการ API เพื่อเชื่อมต่อไปยัง AI โมเดลที่ทาง OpenAI เตรียมไว้เป็นต้น

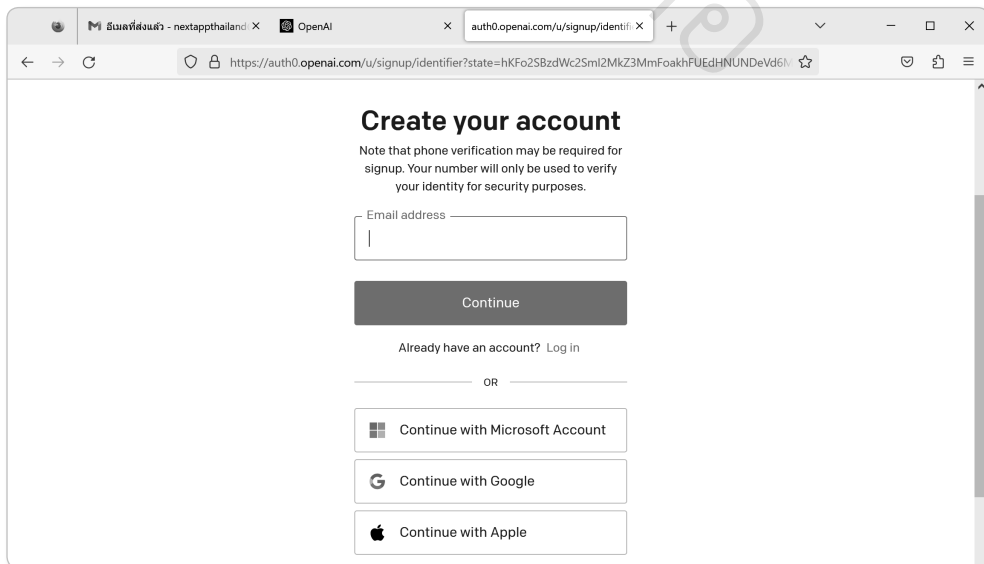
ChatGPT สุดยอด AI จาก OpenAI

ChatGPT คือ ปัญญาประดิษฐ์ (AI) ที่ถูกสร้างขึ้นมาเพื่อใช้สื่อสารกับผู้ใช้ ผ่านรูปแบบของการสนทนา (เรียกว่า Chat) เหมือนกับว่ากำลังแชทพูดคุยอยู่กับคนจริง ๆ โดย ChatGPT จะทำหน้าที่เหมือนผู้ช่วยส่วนตัว ซึ่งเราสามารถถามคำถามหรือพูดคุยเรื่องอะไรก็ได้ (ที่อยู่ในกฎที่กำหนด) โดย ChatGPT จะพยายามตอบคำถามให้เข้าใจได้ง่ายที่สุดเท่าที่ ChatGPT จะตอบได้ หรือเท่าที่มีข้อมูล ณ เวลานั้น ๆ

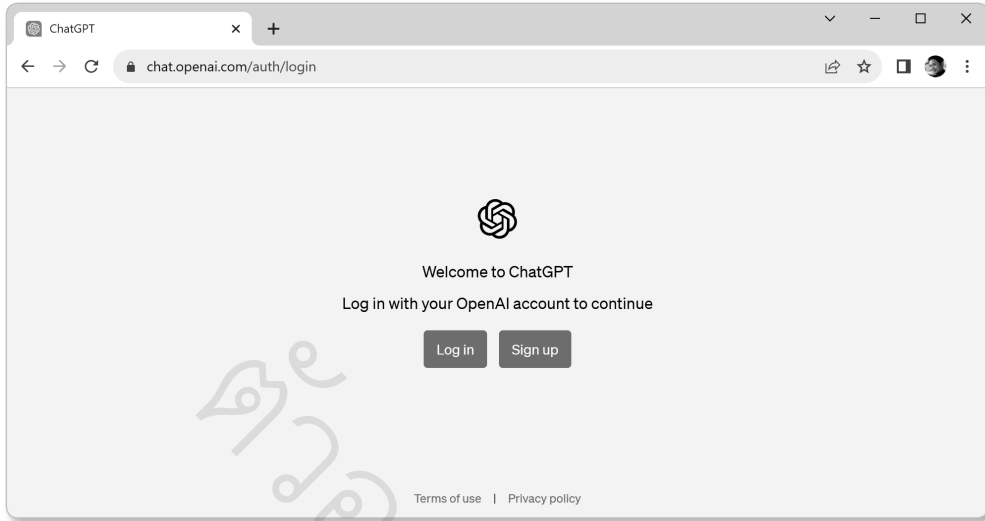
ก่อนใช้งาน ChatGPT จะต้องลงทะเบียนกับ OpenAI โดยไปที่แอดเดรส <https://openai.com> จากนั้นคลิกลิงก์ Sign up เพื่อลงทะเบียน



สามารถลงทะเบียนด้วยอีเมล บัญชี Google, บัญชี Microsoft หรือบัญชี Apple ก็ได้ ในระหว่างขั้นตอนการลงทะเบียน จะต้องกรอกหมายเลขโทรศัพท์เพื่อใช้ยืนยันตัวตน



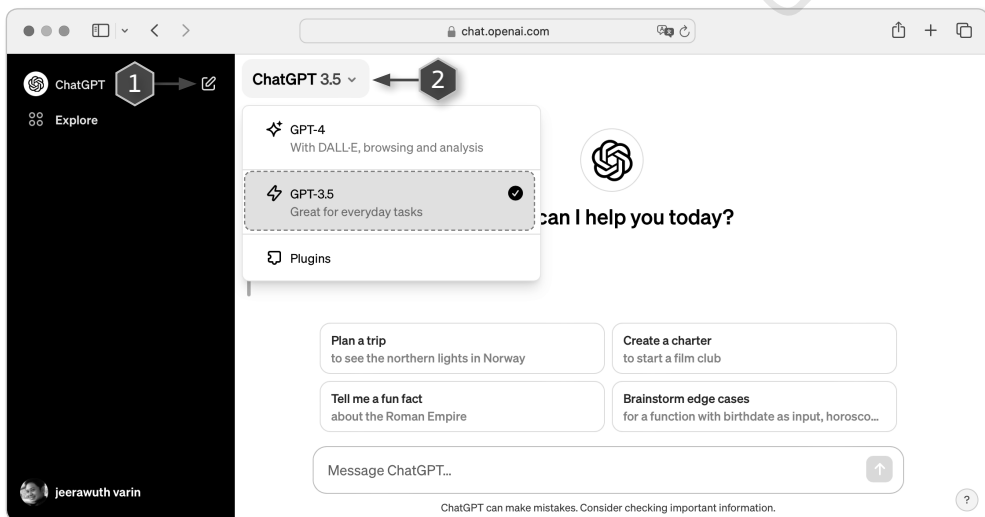
หลังจากลงทะเบียนเรียบร้อยแล้ว ก็ให้เข้าไปยังบริการ ChatGPT ของ OpenAI หรือมายังแอดเดรส <https://chat.openai.com> คลิกปุ่ม Log in หลังจากนั้นจะพบหน้าจอสำหรับใช้พูดคุยกับ ChatGPT ตามต้องการ



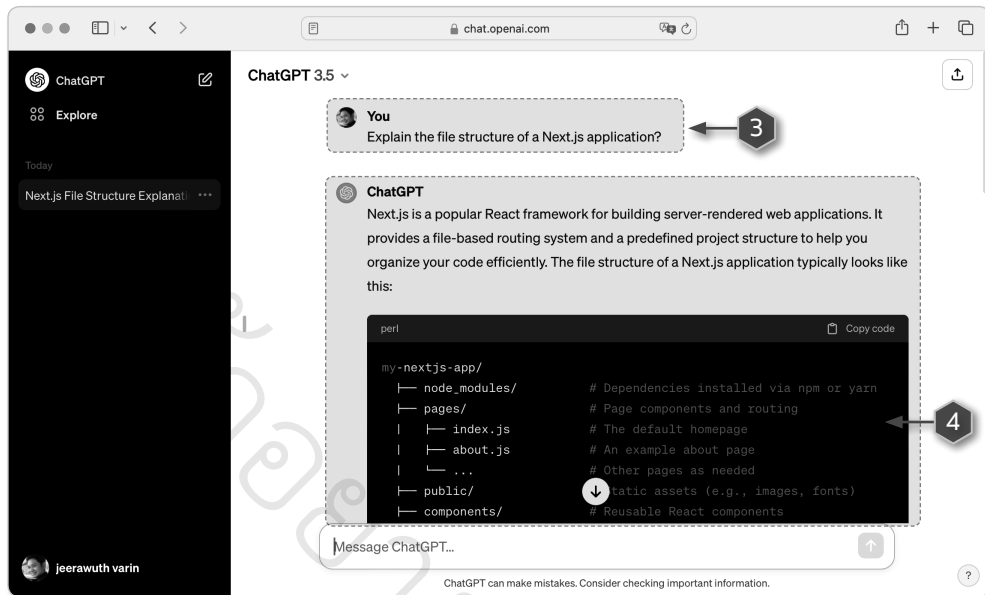
สนทนากับ ChatGPT ด้วยการกำหนด Prompt

การใช้งาน ChatGPT ไม่มีอะไรยุ่งยาก เพียงแค่กรอกข้อความที่เรียกว่า Prompt ลงไป และรอผลลัพธ์จาก ChatGPT ตอบกลับมา โดย Prompt จะมีลักษณะเป็นข้อความทั่ว ๆ ไป คล้ายกับข้อความที่เราใช้แชตคุยกับเพื่อน ดังนั้น Prompt จึงสามารถเป็นข้อความที่มีลักษณะเป็นคำถาม ข้อความที่ใช้ขอคำแนะนำ หรือเป็นข้อความที่ใช้เป็นคำสั่งเพื่อบอกให้ ChatGPT ทำบางอย่างให้ก็ได้

1. คลิกที่ปุ่ม  เพื่อสร้างหัวข้อใหม่สำหรับการสนทนา
2. เลือกโมเดล เช่น GPT-3.5 (สำหรับผู้ทั่วไป) หรือ GPT-4 (สำหรับผู้ที่ใช้เสียค่าบริการเพื่ออัปเกรดเป็น ChatGPT Plus)

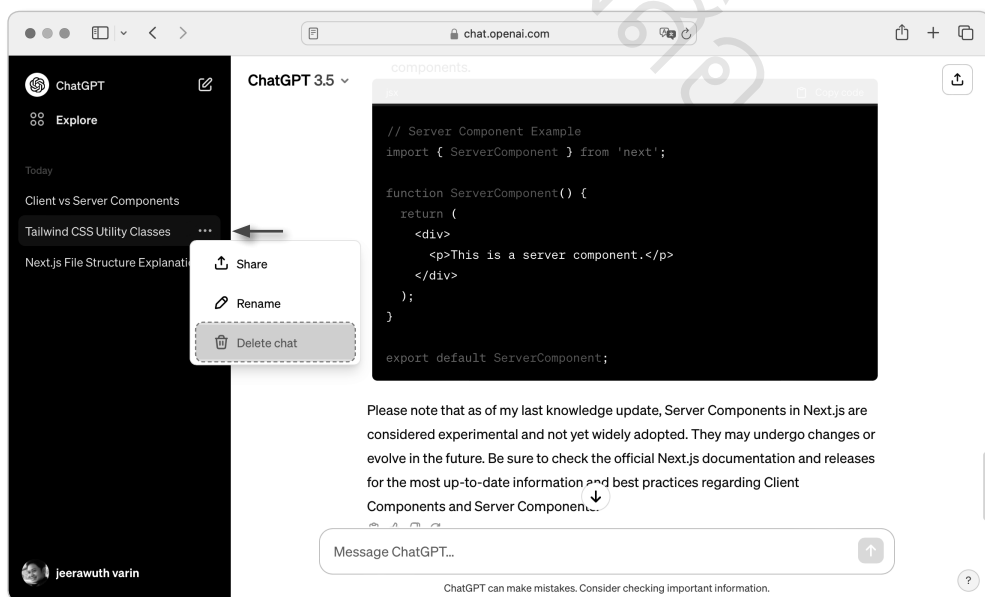


3. กรอกคำถามที่ต้องการ เช่น Explain the file structure of a Next.js application? จากนั้นกดปุ่ม **<Enter>** หรือคลิกปุ่ม **↑**
4. ก็จะได้คำตอบจาก ChatGPT ตามต้องการ



ถ้าต้องการสนทนากับ ChatGPT ด้วยหัวข้อใหม่ แนะนำให้คลิกปุ่ม **✎** เพื่อแยกหัวข้อเดิมกับหัวข้อใหม่ออกจากกัน ซึ่ง ChatGPT จะเก็บประวัติการสนทนาของทั้งสองหัวข้อ ซึ่งเราสามารถย้อนกลับมาดูข้อมูลหรือมาสนทนาต่อในหัวข้อนั้น ๆ ได้

สำหรับหัวข้อไหนไม่ได้ใช้แล้วก็สามารถลบออกไปได้ โดยคลิกปุ่ม **...** และเลือกคำสั่ง Delete chat



การใช้งานปลั๊กอิน

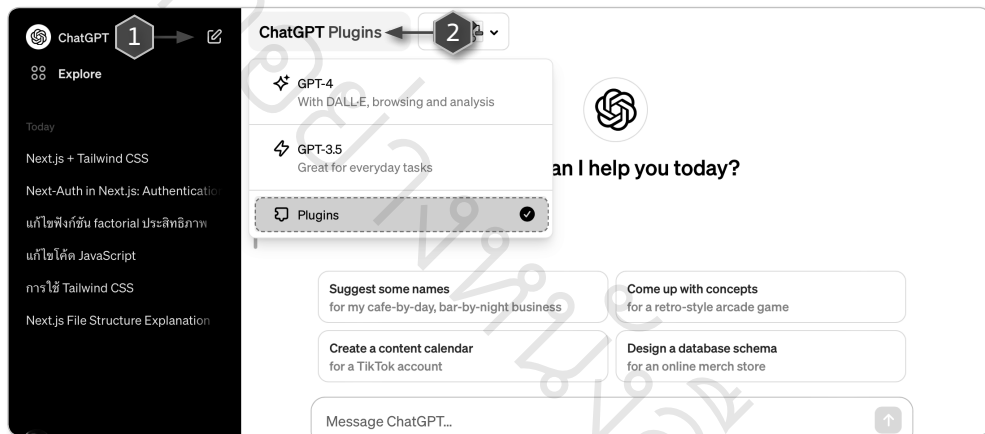
การใช้ปลั๊กอิน (plugins) ใน ChatGPT Plus เป็นการเพิ่มความสามารถใหม่ ๆ ให้กับ ChatGPT ได้ ยกตัวอย่างเช่น ปลั๊กอินสำหรับอ่านและสรุปข้อมูลจากไฟล์ PDF ปลั๊กอินสำหรับแปลงข้อความให้กลายเป็นเสียง ปลั๊กอินสำหรับนำข้อมูลที่ได้ไปสร้างเป็นกราฟ ฯลฯ

การใช้งานปลั๊กอินใน ChatGPT Plus สามารถแบ่งได้เป็น 2 ขั้นตอน ดังนี้

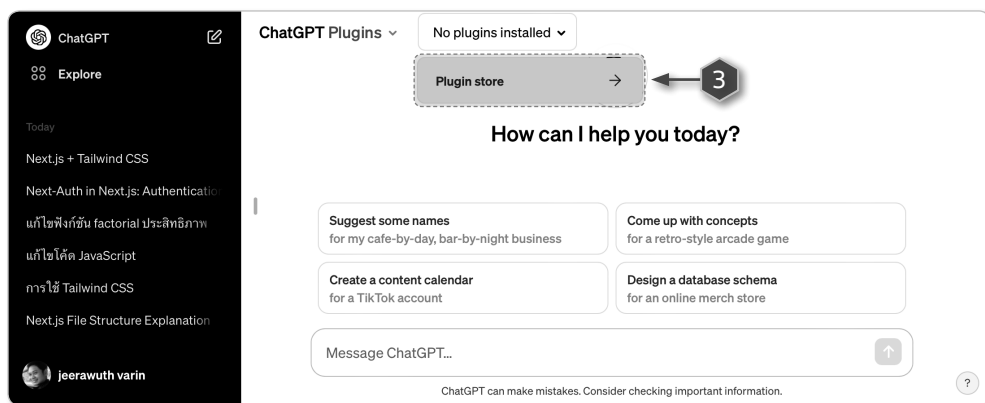
ขั้นตอนที่ 1 ติดตั้งปลั๊กอิน

ติดตั้ง (install) ปลั๊กอินที่ต้องการ โดยสามารถค้นหาปลั๊กอินจาก Plugin Store ของ ChatGPT หลังจากติดตั้งปลั๊กอินไปแล้วสามารถถอดปลั๊กอิน (uninstall) ที่ไม่ต้องการได้

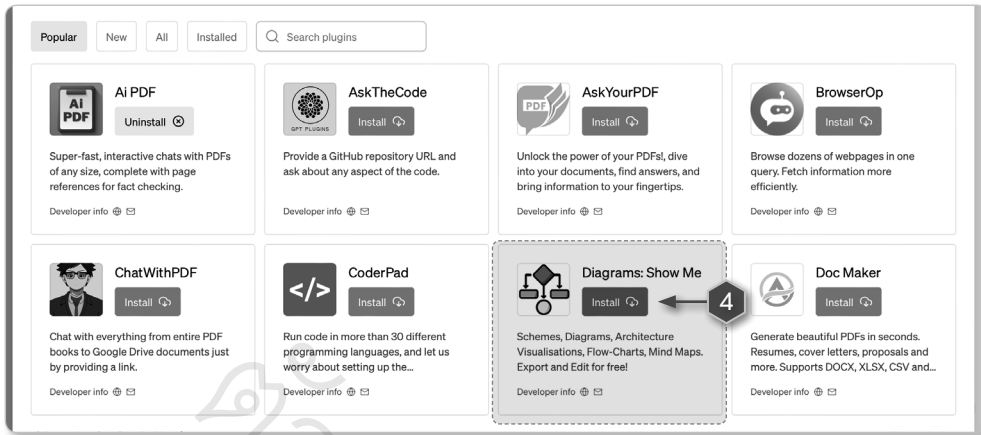
1. คลิก  เพื่อสร้างบทสนทนาใหม่
2. คลิกเลือกเมนู Plugins



3. ในการใช้งานครั้งแรก จะพบข้อความ No plugins installed ให้คลิกที่ Plugin store เพื่อเลือกรายชื่อปลั๊กอินจาก Plugin store



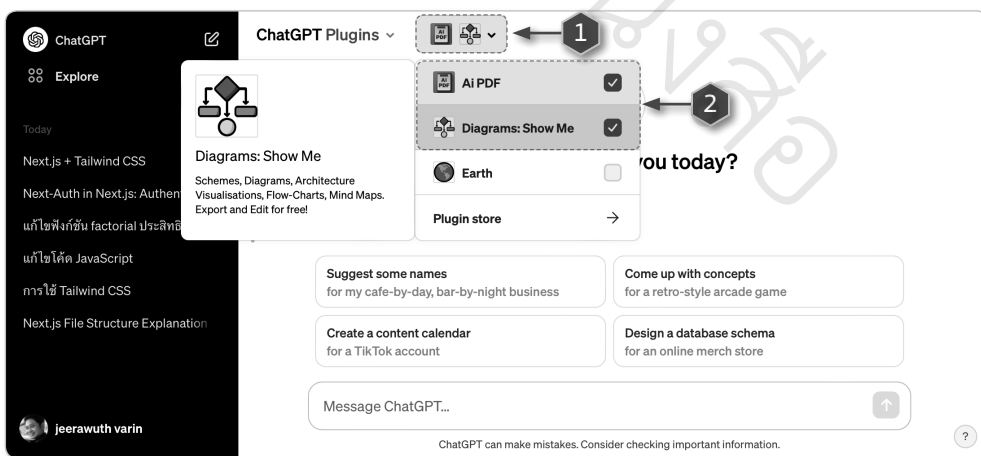
4. คลิกปุ่ม Install เพื่อติดตั้งปลั๊กอินที่ต้องการ ในตัวอย่างติดตั้งปลั๊กอิน Diagrams: Show Me สำหรับใช้วาดไดอะแกรม



ขั้นตอนที่ 2 เลือกใช้งานปลั๊กอิน

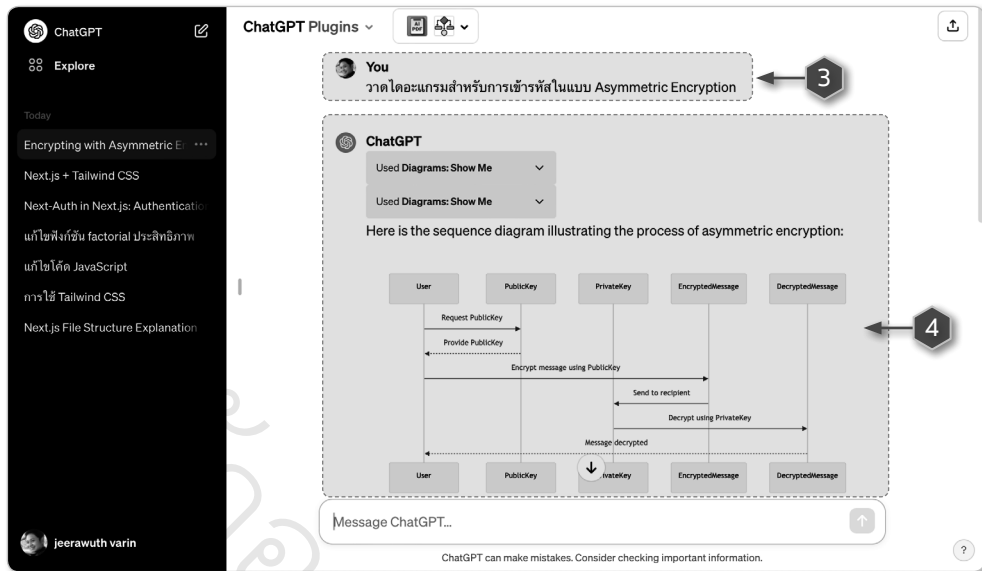
ในการกรอก Prompt เพื่อสั่งให้ ChatGPT นำความสามารถจากปลั๊กอินมาใช้งานนั้น แนะนำให้เลือกเปิดใช้งานเฉพาะปลั๊กอินที่เกี่ยวข้องกับ Prompt นั้น ๆ

1. ให้สังเกตไอคอนแสดงรายการปลั๊กอินที่ได้ถูกติดตั้งและถูกเปิดใช้งาน ให้คลิกปุ่มลูกศรข้างไอคอนปลั๊กอินเพื่อดูรายการปลั๊กอินทั้งหมด
2. คลิกถูกเลือกเฉพาะปลั๊กอินที่ต้องการ (ในตัวอย่างเปิดใช้งานเฉพาะ Diagrams: Show Me และ Ai PDF)



3. กรอก Prompt ซึ่ง ChatGPT จะนำความสามารถที่อยู่ในปลั๊กอินมาใช้งานโดยอัตโนมัติ ตัวอย่างเช่น ต้องการวาดไดอะแกรม โดยใช้ความสามารถจากปลั๊กอิน Diagrams: Show Me จึงกรอก Prompt เป็น “วาดไดอะแกรมสำหรับการเข้ารหัสในแบบ Asymmetric Encryption”

4. ChatGPT จะแสดงข้อปล้กอินที่กำลังใช้งาน รอสักครู่ก็จะได้โดยะแกรมตามต้องการ

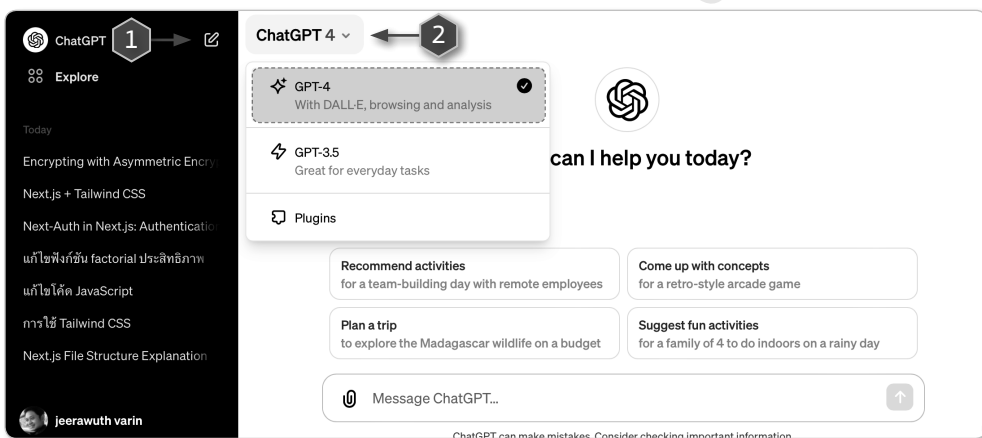


การวิเคราะห์ข้อมูล

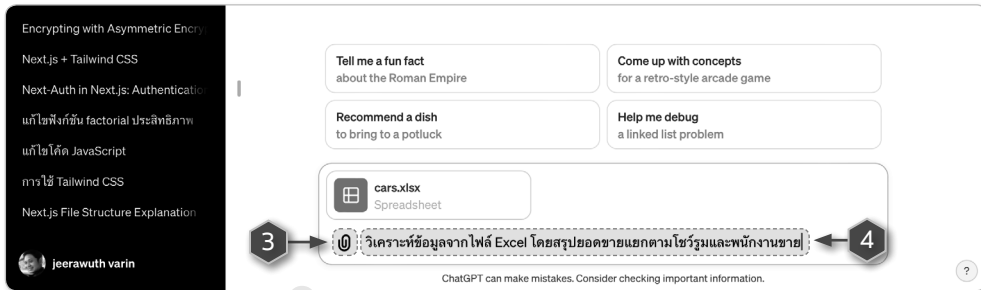
ChatGPT Plus ได้เพิ่มคุณสมบัติใหม่ที่น่าสนใจ คือ ความสามารถในการวิเคราะห์ข้อมูล (Advanced Data Analysis) ตัวอย่างเช่น การวิเคราะห์ข้อมูลการขาย เช่น การวิเคราะห์ยอดขายตามภูมิภาค การเปรียบเทียบยอดขายในแต่ละเดือน ฯลฯ

โดยเริ่มแรกจะต้องมีข้อมูลที่จะใช้วิเคราะห์ก่อน ซึ่งข้อมูลอาจอยู่ในรูปของข้อความ หรือข้อมูลจากไฟล์ต่าง ๆ เช่น ไฟล์ Excel จากนั้นก็ให้กรอก Prompt เพื่อกำหนดรูปแบบในการวิเคราะห์ข้อมูล

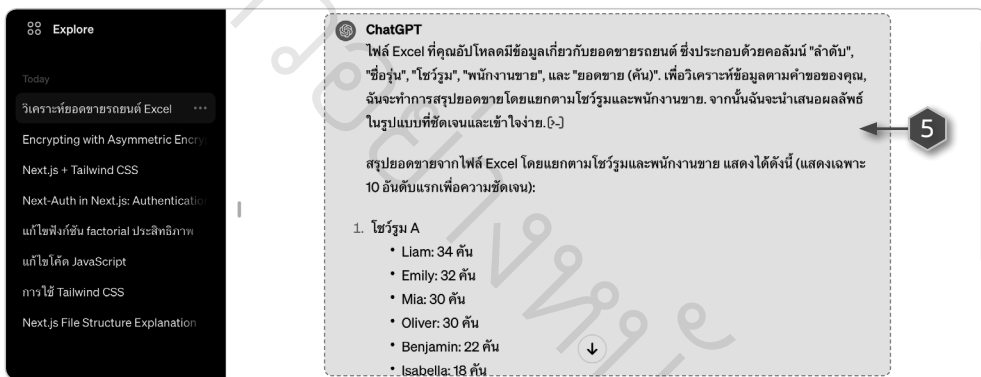
1. คลิก  เพื่อสร้างบทสนทนาใหม่
2. คลิกเมนู GPT-4 เพื่อใช้งาน ChatGPT Plus



3. คลิกปุ่ม  เพื่ออัปโหลดไฟล์ ซึ่งในที่นี้ได้อัปโหลดไฟล์ Excel ที่เก็บข้อมูลการขายสินค้าเอาไว้
4. กรอก Prompt เพื่อวิเคราะห์ข้อมูลการขาย เช่น “วิเคราะห์ข้อมูลจากไฟล์ Excel โดยสรุปยอดขายแยกตามชื่อร้านและพนักงานขาย”



5. ChatGPT จะเข้าไปตรวจสอบไฟล์ Excel และวิเคราะห์ข้อมูลตามที่เรากำลังต้องการ



การใช้ ChatGPT สร้างรูปภาพจาก DALL-E 3

ChatGPT Plus ได้เพิ่มคุณสมบัติการสร้างรูปภาพโดยใช้ DALL-E 3 ซึ่งเราสามารถกรอก Prompt เพื่อใช้สร้างรูปภาพได้ ดังตัวอย่างต่อไปนี้

1. คลิก  เพื่อสร้างบทสนทนาใหม่
2. คลิกเมนู GPT-4

