

Streamlit

ทำเว็บไม่จ้อโค้ดเยอะด้วย Streamlit

อนุชิต ชโลธ

สำนักพิมพ์ทื่อปวง

Streamlit

ทำเว็บไม่ง้อโค้ดเยอะด้วย Streamlit

อนุชิต ชโลธร

คำนำ

ในยุคที่ข้อมูลมีความสำคัญอย่างยิ่ง การนำเสนอข้อมูล, ผลการวิเคราะห์ หรือแม้กระทั่งโมเดล Machine Learning ในรูปแบบที่เข้าถึงง่ายและโต้ตอบได้กลายเป็นทักษะที่จำเป็นสำหรับนักพัฒนา นักวิเคราะห์ข้อมูล และนักวิทยาศาสตร์ข้อมูลจำนวนมาก แต่บ่อยครั้งที่การจะสร้างเว็บแอปพลิเคชันขึ้นมาสักตัวนั้นดูเป็นเรื่องน่าปวดหัว ต้องเรียนรู้ทั้ง Frontend (HTML, CSS, JavaScript) และ Backend ซึ่งอาจใช้เวลานานและมีความซับซ้อนสูง โดยเฉพาะสำหรับผู้ที่ยังเชี่ยวชาญ Python และต้องการเพียงเครื่องมือง่ายๆ ในการ “โซว์ของ” ที่ตัวเองสร้างขึ้น

Streamlit เป็นเฟรมเวิร์ก Python แบบโอเพนซอร์สที่ปฏิวัติวิธีการสร้างเว็บแอปพลิเคชันสำหรับงานด้านข้อมูล มันถูกออกแบบมาเพื่อให้ นักพัฒนา Python สามารถเปลี่ยนสคริปต์ข้อมูลให้กลายเป็นเว็บแอปฯ ที่สวยงามและโต้ตอบได้ภายในเวลาไม่กี่นาที โดยแทบไม่ต้องแตะต้องโค้ด Frontend เลย คุณจะเห็นว่าโค้ด Python ไม่กี่บรรทัด คุณก็สามารถสร้างองค์ประกอบต่างๆ บนหน้าเว็บ รับ Input จากผู้ใช้ แสดงกราฟ และโต้ตอบกับข้อมูลได้อย่างน่าทึ่ง

หนังสือเล่มนี้เหมาะสำหรับใคร

- **นักเรียน นักศึกษา และผู้เริ่มต้นเขียน Python** ที่ต้องการนำเสนอโปรเจกต์หรือข้อมูลในรูปแบบที่น่าสนใจและเข้าถึงง่าย
- **นักวิเคราะห์ข้อมูล (Data Analysts)** ที่ฝึนอยากจะสร้าง Dashboard แบบ Interactive เพื่อแชร์ Insight ให้กับทีมหรือผู้บริหาร โดยไม่ต้องรอทีมพัฒนาเว็บ
- **นักวิทยาศาสตร์ข้อมูล (Data Scientists)** ที่ต้องการนำโมเดล Machine Learning มาให้ผู้ใช้งานทดลองใช้งานได้อย่างรวดเร็ว
- **โปรแกรมเมอร์ Python ทุกระดับ** ที่อยากมีเครื่องมือสร้างเว็บแอปฯ ติดตัวไว้สำหรับโปรเจกต์ส่วนตัวหรืองาน Prototype อย่างเร่งด่วน
- **ทุกคนที่สนใจสร้างเว็บแอปพลิเคชัน** โดยเน้นความเรียบง่าย รวดเร็ว และใช้ Python เป็นหลัก

สิ่งที่คุณจะได้เรียนรู้จากหนังสือเล่มนี้

เราจะเริ่มต้นเดินทางกันตั้งแต่ทำความรู้จักว่า Streamlit คืออะไร และทำไมมันถึงเป็นเครื่องมือที่ทรงพลัง จากนั้นจะพาคุณติดตั้งและสร้าง “Hello, Streamlit!” แอปแรกของคุณอย่างรวดเร็ว คุณจะได้เรียนรู้การใช้งาน Widgets ต่างๆ เพื่อรับ Input จากผู้ใช้ ไม่ว่าจะเป็นปุ่ม, ช่องกรอกข้อความ, ตัวเลือกแบบต่างๆ ไปจนถึงการอัปโหลดไฟล์

ในส่วนของการแสดงผล เราจะสำรวจวิธีการนำเสนอข้อมูลในรูปแบบตาราง, JSON, การแสดงรูปภาพ, เสียง, วิดีโอ และที่สำคัญคือการเนรมิตกราฟสวยๆ จากไลบรารียอดนิยมอย่าง Matplotlib, Plotly และ Altair นอกจากนี้ คุณจะได้เรียนรู้วิธีการจัดระเบียบหน้าเว็บของคุณให้สวยงามและใช้งานง่ายด้วย Layout และ Containers ต่างๆ เช่น Sidebar, Columns และ Tabs

เมื่อคุณคุ้นเคยกับพื้นฐานแล้ว เราจะก้าวไปอีกขั้นกับการจัดการสถานะ (Session State) เพื่อสร้างแอปพลิเคชันที่มีความต่อเนื่อง, การทำ Caching เพื่อเพิ่มประสิทธิภาพให้แอปฯ ของคุณทำงานได้รวดเร็วยิ่งขึ้น และการสร้างแอปพลิเคชันแบบหลายหน้า (Multipage Apps) เพื่อจัดการโปรเจกต์ที่ซับซ้อนขึ้น

ท้ายที่สุด คุณจะได้ลงมือทำโปรเจกต์ตัวอย่างจริงหลายโปรเจกต์ ตั้งแต่การสร้าง Dashboard วิเคราะห์ข้อมูล ไปจนถึงการสร้างเว็บแช

ทบทวน เบื้องต้น เพื่อให้เห็นภาพรวมของการนำความรู้ทั้งหมดมาประยุกต์ใช้ และแน่นอน เราจะพูดถึงขั้นตอนสำคัญอย่างการ Deploy แอปพลิเคชันของคุณขึ้นสู่โลกออนไลน์ผ่าน Streamlit Community Cloud และแพลตฟอร์มอื่นๆ เพื่อให้ผลงานของคุณสามารถเข้าถึงได้จากทุกคน

ผมหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะเป็นคู่มือที่ช่วยให้คุณค้นพบความสนุกและความสะดวกในการสร้างเว็บแอปพลิเคชันด้วย Python และ Streamlit เปิดประตูสู่ความเป็นไปได้ใหม่ๆ ในการนำเสนอข้อมูล และโปรเจกต์ของคุณ ขอให้สนุกกับการเดินทาง “ทำเว็บไม่จ้อโค้ดเยอะ” ไปด้วยกัน!

สารบัญ

Streamlit คืออะไร? ทำไมต้องใช้?	1
ปัญหาการนำเสนอข้อมูลด้วย Python	2
ปรัชญาและความสามารถหลักของ Streamlit	3
ข้อดีของ Streamlit	5
Streamlit เหมาะกับใครและงานแบบไหน	6
ติดตั้งและเริ่มต้นโปรเจกต์แรก	9
สิ่งที่ต้องมี	9
ขั้นตอนการติดตั้ง	10
สร้างไฟล์แรกของคุณ	10
รันแอป Streamlit	11
ทำความเข้าใจโครงสร้างพื้นฐาน	12
ลองแสดงข้อความง่ายๆ	13
รู้จักกับ Widgets พื้นฐาน	15
Widgets คืออะไร? ทำไมถึงสำคัญ?	15
การสร้างปุ่ม (Button) และการตอบสนอง	16
การรับข้อความ (Text Input)	17
การรับตัวเลข (Number Input)	19
การเลือก (Selectbox, Multiselect)	20

ตัวเลือกแบบ Checkbox และ Radio button	22
Slider และ Select Slider	24
การรับวันที่และเวลา	26
การอัปโหลดไฟล์	28
Workshop สร้างฟอร์มรับข้อมูลพื้นฐาน	30
การแสดงผลข้อมูล	34
การแสดง DataFrames (Pandas)	34
การแสดงผลข้อมูล JSON	36
การแสดงผลรูปภาพ, เสียง และ วิดีโอ	37
การแสดงผลค่าตัวเลขในรูปแบบ Metric และ KPIs	40
การแจ้งเตือนและสถานะ	41
Progress bar และ Spinner	43
Workshop สร้างหน้าแสดงผลข้อมูลจากไฟล์ CSV	45
เนรมิตกราฟสวยด้วย Streamlit	50
การแสดงกราฟ	50
การแสดงกราฟด้วย Matplotlib	51
การแสดงกราฟด้วย Altair	53
การแสดงกราฟด้วย Plotly	56
การแสดงกราฟด้วย Vega-Lite	58
การแสดงผลข้อมูลบนแผนที่ด้วย PyDeck	61
เคล็ดลับการสร้างกราฟแบบ Interactive	63

Workshop การสร้าง Dashboard แสดงกราฟจากข้อมูล	65
จัดระเบียบ Layout ให้สวยงาม	71
การจัดวางองค์ประกอบในแนวตั้ง	71
การใช้ Sidebar	72
การจัดวางองค์ประกอบในแนวนอน	73
การใช้ Expander	75
การใช้ Containers	77
การใช้ Tabs	79
การจองพื้นที่เพื่ออัปเดตภายหลัง	80
Workshop ออกแบบหน้าเว็บที่มี Sidebar และแบ่งคอลัมน์ แสดงข้อมูล	82
การจัดการสถานะในแอปพลิเคชัน	88
ทำความเข้าใจการ Rerun ของ Streamlit	88
Session State คืออะไร	89
การเก็บและเรียกใช้ค่าใน Session State	89
ตัวอย่างการจำค่าที่ผู้ใช้กรอกและการสร้างแอปหลายขั้นตอน	91
การจำค่าที่ผู้ใช้กรอก (Remembering User Input)	92
การสร้างแอปพลิเคชันหลายขั้นตอน (Multi-step App)	93
Callback functions สำหรับ Widgets	96
Workshop สร้างแอปพลิเคชันนับเลข หรือ Quiz ที่ต้องจำ สถานะ	99

Workshop แอปตัวนับเลข	100
Workshop แอป Quiz จดจำคะแนน	101
การ Caching เพื่อเพิ่มประสิทธิภาพ	107
ทำไม Caching ถึงสำคัญสำหรับแอป Streamlit?	107
@st.cache_data สำหรับการ Cache ข้อมูลที่ Return ค่า	108
@st.cache_resource สำหรับการ Cache ทรัพยากรที่ไม่สามารถ Serialize ได้	112
Workshop สร้างแอปพลิเคชันวิเคราะห์ข้อมูลพร้อม Caching	116
การสร้างแอปพลิเคชันแบบหลายหน้า (Multipage Apps)	124
โครงสร้างโฟลเดอร์สำหรับแอปหลายหน้า (โฟลเดอร์ pages)	124
การสร้างหน้าย่อยต่างๆ	125
การเชื่อมโยงระหว่างหน้า	127
Workshop แปลงโปรเจกต์เดิมให้เป็นแอปพลิเคชันหลายหน้า	129
การปรับแต่งธีม	136
การตั้งค่า Theme	136
การใช้ HTML และ CSS เบื้องต้น	138
ข้อควรระวังในการใช้ HTML/CSS โดยตรง	136
การใช้ Components สำเร็จรูป	136
การนำแอปพลิเคชันของคุณไปใช้งานจริง	140
ใช้ Streamlit Cloud ง่ายและสะดวก	140
Streamlit Cloud คืออะไร	140

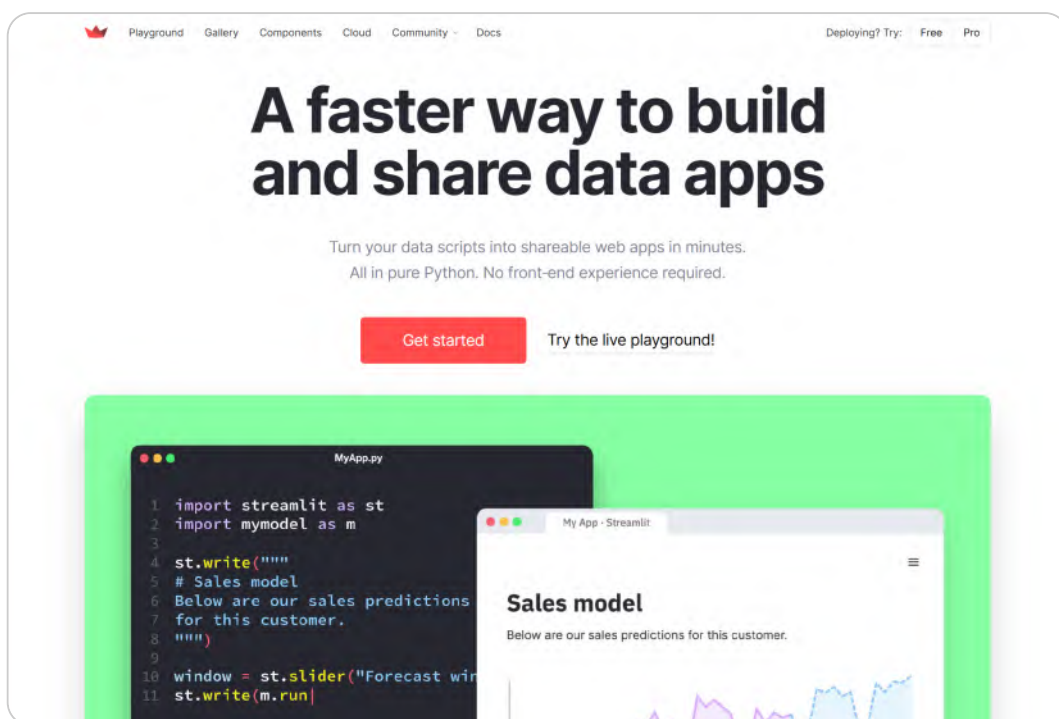
เตรียมโปรเจกต์ให้พร้อมก่อน	140
อัปโหลดโค้ดขึ้น GitHub	142
เริ่ม Deploy แอปขึ้น Streamlit Cloud	142
ใช้ Docker ยืดหยุ่นและควบคุมได้มากขึ้น	143
เขียน Dockerfile (ใบสั่งสร้างบ้านให้แอป)	143
รันแอปด้วย Docker	144
(ทางเลือก) ใช้ Docker Compose (ตัวช่วยจัดการ Docker)	145
ตั้งค่า SSL/HTTPS (เพิ่มความปลอดภัยและความน่าเชื่อถือ)	147
ด้วย Nginx และ Let's Encrypt	
ติดตั้ง Nginx บนเซิร์ฟเวอร์	147
ตั้งค่า Nginx เป็น Reverse Proxy	147
ติดตั้ง Certbot และขอใบรับรอง SSL	149
ตัวอย่างไฟล์ตั้งค่า Nginx ที่มี SSL แล้ว (Certbot ทำให้)	149
ทดสอบ SSL	151
ตั้งค่าให้ต่ออายุ SSL อัตโนมัติ	151
เลือกวิธีไหนดี?	152
เคล็ดลับเพิ่มเติม แสดงเวอร์ชันของแอปจาก Git	152
สร้าง Dashboard วิเคราะห์ข้อมูลอย่างง่าย	154
โจทย์ (Problem Statement)	154
เตรียมข้อมูล (Data Preparation)	155
ขั้นตอนการสร้าง Dashboard	156

โหลดข้อมูลและแสดงตาราง	156
แสดงตัวเลขสำคัญ	157
สร้างกราฟแนวโน้มยอดขาย	158
สร้างกราฟยอดขายตามหมวดหมู่	159
เพิ่ม Filter ใน Sidebar	160
รวมโค้ดทั้งหมด	163
อธิบายโค้ด	166
แนวทางการต่อยอด	170
สรุป Workshop นี้	170
สร้าง Chatbot ด้วย Streamlit และ Google Gemini	172
การเตรียมความพร้อม	172
ติดตั้ง Library ที่จำเป็น	172
Google Gemini API Key	172
ตั้งค่า Secrets File	172
โครงสร้างโค้ด	173
Import Libraries และตั้งชื่อแอป	173
การตั้งค่า Google Gemini Client	173
การจัดการ Session State	174
แสดงประวัติการสนทนา	176
รับ Input และประมวลผล	176
พีเจอร์เสริม	178

วิธีการรันแอป	179
การปรับแต่งเพิ่มเติม	179
เพิ่มการตั้งค่าโมเดล	180
เพิ่มการจำกัดจำนวนข้อความ	180
เพิ่มการส่งออกประวัติ	180
ข้อควรระวัง	181
สรุป Workshop นี้	181
ดาวนโหลดซอร์สโค้ด	182

Streamlit คืออะไร? ทำไมต้องใช้?

ยินดีต้อนรับสู่บทแรกของเส้นทางการสร้างเว็บแอปพลิเคชันด้วย Python ที่ง่ายและรวดเร็วที่สุด ก่อนที่เราจะลงลึกถึงวิธีการเขียนโค้ดและเนรมิตไอเดียของคุณให้กลายเป็นเว็บแอปฯ ที่จับต้องได้ เรามาทำความรู้จักกับพระเอกของเรากันก่อน นั่นคือ **Streamlit**



Streamlit

บางทีคุณอาจกำลังคิดว่า “ฉันก็เขียน Python ได้อยู่แล้ว ทำไมต้องมาเรียนรู้อะไรเพิ่มอีก?” หรือ “การสร้างเว็บมันน่าจะยุ่งยาก ต้องรู้ทั้ง HTML, CSS, JavaScript ไม่ใช่เหรอ?” คำถามเหล่านี้เป็นเรื่องปกติ

และในบทนี้ เราจะมาดูกันว่า Streamlit จะเข้ามาตอบโจทย์และเปลี่ยนความคิดของคุณเกี่ยวกับการสร้างเว็บแอปพลิเคชันได้อย่างไร

ปัญหาการนำเสนอข้อมูลด้วย Python

ลองจินตนาการว่าคุณเป็น...

- **นักวิเคราะห์ข้อมูล** ที่เพิ่งค้นพบ Insight เจ๋งๆ จากชุดข้อมูลขนาดใหญ่ คุณพล็อตกราฟสวยด้วย Matplotlib หรือ Seaborn บน Jupyter Notebook และอยากแชร์ให้ทีมหรือผู้บริหารดู แต่จะให้ทุกคนมารัน Notebook เอง หรือส่งแค่ภาพนิ่งของกราฟไป มันก็ไม่ตอบโจทย์ โดยเฉพาะถ้าคุณอยากให้ลองปรับพารามิเตอร์ แล้วดูผลแบบเรียลไทม์
- **นักวิทยาศาสตร์ข้อมูล** ที่พัฒนาโมเดล Machine Learning สำเร็จ เช่น โมเดลทำนายราคาบ้าน หรือจำแนกรูปภาพ และอยากให้คนอื่นทดลองใช้ แต่ติดที่ว่าต้องติดตั้ง Python และ Library เยอะๆ หรือไม่ก็ต้องหานักพัฒนาเว็บมาทำ Interface ให้ ซึ่งทั้งยุ่งยากและใช้ทรัพยากร
- **นักพัฒนา Python** ที่มีไอเดียสร้างเครื่องมือเล็กๆ อย่างเว็บคำนวณ เว็บแปลงหน่วย หรือแสดงข้อมูลจาก API แต่พอคิดว่าจะต้องทำ Frontend, Backend เองทั้งหมด ก็เริ่มถอดใจ

ทั้งหมดนี้คือปัญหาที่คนทำงานสาย Python เจอบ่อย – การนำเสนอผลงานให้ *โต้ตอบได้* และ *เข้าถึงง่าย* นั้นไม่ใช่เรื่องง่าย ต้องใช้ทักษะหลายด้าน ใช้เวลามาก และบางครั้งโอเคทีเดียว ก็ถูกพับเก็บไว้เพราะข้อจำกัดเหล่านี้

ปรัชญาและความสามารถหลักของ Streamlit

Streamlit (สตรีม-ลิต) เป็นเฟรมเวิร์ก Python แบบโอเพนซอร์ส สร้างขึ้นมาเพื่อแก้ปัญหานี้โดยเฉพาะ ปรัชญาหลักของ Streamlit เรียบง่ายมาก **“The fastest way to build and share data apps.”** หรือ “วิธีที่เร็วที่สุดในการสร้างและแชร์เว็บแอปพลิเคชันข้อมูล”

หัวใจสำคัญของ Streamlit คือการช่วยให้คุณสามารถเปลี่ยนสคริปต์ Python ที่คุณเขียนสำหรับวิเคราะห์ข้อมูลหรือรันโมเดล ให้กลายเป็นเว็บแอปพลิเคชันเต็มรูปแบบได้โดยแทบไม่ต้องมีความรู้เรื่องการพัฒนาเว็บแบบดั้งเดิม (HTML, CSS, JavaScript) เลย คุณเขียนโค้ด Python เหมือนเดิม เพิ่มคำสั่งของ Streamlit เข้าไปเพียงเล็กน้อย แอปของคุณก็พร้อมที่จะแสดงผลบนเว็บเบราว์เซอร์ และให้ผู้ใช้โต้ตอบได้ทันที

ความสามารถหลักของ Streamlit ได้แก่

- **การแสดงผลข้อมูล** แสดงข้อความ, ตารางข้อมูล (เช่น Pandas DataFrame), รูปภาพ, เสียง, วิดีโอ ได้อย่างง่ายดาย
- **วิดเจ็ต (Widgets) แบบ Interactive** สร้างปุ่ม, สไลเดอร์, ช่องรับข้อความ, เมนูตัวเลือก, ช่องอัปโหลดไฟล์ ฯลฯ เพื่อให้ผู้ใช้สามารถป้อนข้อมูลหรือควบคุมการทำงานของแอปได้
- **การพล็อตกราฟ** รองรับไลบรารีกราฟยอดนิยมของ Python เช่น Matplotlib, Seaborn, Plotly, Altair, Vega-Lite ทำให้คุณนำกราฟสวยๆ ที่สร้างไว้อยู่แล้วมาแสดงบนเว็บได้ทันที
- **Layout ที่ยืดหยุ่น** จัดระเบียบองค์ประกอบต่างๆ บนหน้าเว็บได้ง่ายๆ เช่น การสร้าง Sidebar, การแบ่งคอลัมน์
- **Caching อัจฉริยะ** ช่วยให้แอปของคุณทำงานเร็วขึ้นโดยการเก็บผลลัพธ์ของการคำนวณที่ใช้เวลานานไว้
- **การทำงานร่วมกับ Ecosystem ของ Python** เนื่องจาก Streamlit เป็น Python คุณจึงสามารถใช้ไลบรารี Python ที่คุณชื่นชอบได้ทั้งหมด ไม่ว่าจะเป็น NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch และอื่นๆ อีกมากมาย

ลองนึกภาพว่าคุณมีสคริปต์ Python ที่โหลดข้อมูล, ประมวลผล, แล้วพล็อตกราฟอยู่แล้ว คุณเพียงแค่เพิ่ม `import streamlit as st` เข้าไป จากนั้นเปลี่ยนคำสั่ง `print()` เป็น `st.write()` หรือ `plt.show()` เป็น `st.pyplot()` เพียงเท่านี้คุณก็มีเว็บแอปฯ ที่แสดงผลลัพธ์เหล่านั้นได้แล้ว! มันง่ายขนาดนั้นจริงๆ

ข้อดีของ Streamlit

ทำไม Streamlit ถึงได้รับความนิยมอย่างรวดเร็วและกลายเป็นเครื่องมือโปรดของใครหลายคน? นี่คือเหตุผลหลักๆ

1. **ง่าย (Simplicity)** นี่คือการจุดแข็งที่สุดของ Streamlit โค้ดที่ใช้สั้น กระชับ และเข้าใจง่าย คุณไม่จำเป็นต้องเรียนรู้แนวทางการพัฒนาเว็บที่ซับซ้อน Streamlit จัดการเรื่องยากๆ เบื้องหลังให้คุณหมดแล้ว ถ้าคุณเขียน Python ได้ คุณก็สร้างเว็บแอปฯ ด้วย Streamlit ได้
2. **เร็ว (Speed)** คุณสามารถสร้างต้นแบบ (Prototype) ของเว็บแอปฯ ได้ภายในเวลาไม่กี่นาที หรือไม่กี่ชั่วโมง จากสคริปต์ข้อมูลที่คุณมีอยู่แล้ว การแก้ไขและเห็นผลลัพธ์ที่เปลี่ยนแปลงไปบนหน้าเว็บก็ทำได้อย่างรวดเร็ว (Live Reload) ช่วยให้งานการพัฒนาสั้นลงอย่างมาก
3. **Python ล้วนๆ (Pure Python)** ไม่ต้องสลับไปมาระหว่างภาษา Python, HTML, CSS และ JavaScript อีกต่อไป ทุกอย่างเกิดขึ้นในโลกของ Python ที่คุณคุ้นเคย ทำให้คุณโฟกัสไปที่การแก้ปัญหาด้วยข้อมูลและตรรกะของ Python ได้เต็มที่
4. **Community ที่ยอดเยี่ยม (Great Community)** Streamlit มีชุมชนผู้ใช้งานที่เติบโตอย่างรวดเร็วและพร้อมให้ความช่วยเหลือ มีเอกสารประกอบ (Documentation) ที่ดีเยี่ยม, ฟอรัมสำหรับถามตอบปัญหา และมีตัวอย่างโปรเจกต์เจ๋งๆ ให้ศึกษามากมาย

นอกจากนี้ยังมี Streamlit Components ที่ผู้ใช้สร้างขึ้นมาแบ่งปันเพื่อเพิ่มความสามารถให้กับแอปของคุณได้อีกด้วย

5. **สวยงามพร้อมใช้ (Beautiful by Default)** แอปที่สร้างด้วย Streamlit จะมีหน้าตาที่ดูดีและทันสมัยมาตั้งแต่ต้น โดยที่คุณแทบไม่ต้องปรับแต่งอะไรเลย ทำให้คุณมั่นใจได้ว่าผลงานที่ออกมาจะดูเป็นมืออาชีพ
6. **แชร์ง่าย (Easy to Share)** Streamlit มีบริการ **Streamlit Community Cloud** ที่ให้คุณสามารถ Deploy หรือนำแอปของคุณขึ้นไปโฮสต์บนอินเทอร์เน็ตได้ฟรีและง่ายดาย ทำให้คนทั่วโลกสามารถเข้าถึงและใช้งานแอปของคุณได้

Streamlit เหมาะกับใครและงานแบบไหน

จากที่กล่าวมาทั้งหมด คุณคงพอจะเห็นภาพแล้วว่า Streamlit นั้นมีประโยชน์อย่างยิ่งสำหรับ

- **นักเรียน/นักศึกษา** ที่ต้องการนำเสนอโปรเจกต์ Python, งานวิจัย หรือการบ้านในรูปแบบที่น่าสนใจและโต้ตอบได้ แทนที่จะส่งแค่โค้ดหรือสไลด์แบบเดิมๆ
- **นักวิเคราะห์ข้อมูล (Data Analysts)** ที่ต้องการสร้าง Dashboard, รายงานแบบ Interactive หรือเครื่องมือสำรวจข้อมูล (Data Exploration Tools) เพื่อให้ทีมหรือผู้บริหาร