06026240 INTELLIGENT SYSTEM DEVELOPMENT

Chapter I

Introduction To Intelligent System Development

with Dr. Bhattarabhorn Wattanacheep

15-WEEK COURSE SCHEDULE · Intelligent System Development · Undergraduate · 3+ hrs/session · Lecture + Lab

Category Key: Foundation, Core AI/ML, LLMs & APIs, Engineering, Capstone

Wk	Topic	Theory / Lecture	Lab Activity	Project Milestone	Checkpoint
1	Intro + Git Setup	AI/ML/DL hierarchy · AI stack · Git	Env setup · API call · git init	จัดกลุ่ม + ตั้ง repo เลือก domain	—29/6
2	Computer Vision & AI	Vision basics · Classical CV · CNN	Augmentation and labeling Dataset project (ของอ.)	--	--6/7
3	ML & DL Foundations CNN	Sup/Unsup/RL · Gradient · MLP	Dataset Lab 2 มา Run model	--	--13/7
4	How to build CNN/ RNN	How to build CNN/ RNN	Augmentation and labeling Dataset project (challenge)	ยืนยัน Dataset	✅ Checkpoint 1 20/7
5	Generative Modeling	Autoencoder · VAE · GAN · Diff	Dataset Lab 4 มา Run model	แยกกัน Run model	— 27/7
6	Deep Reinforcement Learning	RL · Q-learning · DQN · policy	คิดคำถาม / คำแนะนำ / ค่าที่จะ evaluate ทั้งหมด	ส่ง Proposal	📄 Proposal Due 3/8
7	Agentic AI & LLMs	Agentic AI & LLMs	LLMOCR extract document / Algorithm เปรียบเทียบท่วงท่า	เพิ่ม LLM เข้า project	—10/8
			Midterm เนื้อหา Wk 1-6 (17/8 เช้า ต้องดูตารางสอบอีกทีนะ)		
8	Transfer & Fine-tune	pre-trained · fine-tune	LLMOCR extract document / Algorithm เปรียบเทียบท่วงท่า		—24/8
			หยุด 31/8 เตรียมลาดกระบังนิทรรศ + open house		
9	Evaluation & Overfit	bias-variance · metrics · reg	Run model ที่คาดว่าจะใช้ใน project ให้ครบ loop	ประเมิน metrics ทุก model	✅ Checkpoint 2 7/9
10	AI APIs & Integration	API · embeddings · function call	ทำโมเดลให้เก่งขึ้น และเลือกโมเดลในกลุ่ม	Model ทำงานได้ ครบ loop	🔌 Integration spike 14/9
11	Frontend Development	HTML/CSS · JS · responsive	Index.html	wireframe + เชื่อม UI	—21/9
12	Backend Development	REST · FastAPI · auth · DB	ทั้งระบบเชื่อมต่อกัน	End to end	✅ Checkpoint 3 28/9
13	สรุปภาพรวม Project	สรุป Project + ทดสอบ Challenge	สรุป Project + ทดสอบ Challenge	--	—ทดสอบ challenge 5/10
14	ทำเอกสาร + ทำ presentation	ทำเอกสาร + presentation	ทำเอกสาร + presentation (อันนี้เดี๋ยว vote : online/onsite)	🔒 Final code freeze	🚫 Code Freeze 12/10
15	Final Presentations	present + demo · review · Q&A	Live demo full-stack AI	ส่ง Code + Slides + Demo	Presentation Day 19/10

GRADE

- Attend class 5% (15 ครั้ง)
- Lab 5% (15 ครั้ง)
- Assignment 30%
- Midterm examination 30% (ปรนัย)
- Final examination 30% (ปรนัย)

การตัดเกรด ตัดตามเกณฑ์ (ช่วงคะแนนอาจมีการปรับเปลี่ยน)

เกรด	ช่วงคะแนน
A	80 - 100
B+	75 - 79
B	70 - 74
C+	65 - 69
C	60 - 64
D+	55 - 59
D	50 - 54
F	00 - 49

ไอเดียที่ใช้ได้จริง

1

OCR Transcript (สจล.)

- ข้อมูล: Synthetic + Synthetic & noise + Real
- ทำ OCR ด้วย CV หรือ LLM-OCR
- ออกแบบ Database สำหรับจัดเก็บข้อมูล
- ยิ่งความแม่นยำสูง ยิ่งได้คะแนนมาก

เน้น OCR ของ document ที่หลากหลาย

2

OCR เล่มหลักสูตร เพื่อตอบคำถามเกี่ยวกับการเรียน

- ทำ OCR (คะแนนส่วนนี้น้อยกว่าโปรเจกต์ 1)
- ใช้ LLM ตอบคำถามระดับง่าย-ยาก จากหลักสูตร
- Database / RAG เก็บข้อมูลทั้งหมดที่จำเป็นต่อการตอบคำถาม

เน้น OCR + LLM + RAG

3

Recommendation การต่อยมวย

- เปรียบเทียบผู้เล่นกับครูฝึก / นักมวย
- เน้น Keypoint / Skeleton Extraction
- สร้างระบบ Recommendation ให้คำแนะนำ
- DB เก็บข้อมูลเล็กน้อย เช่น ท่าครูฝึก เป็นต้น

เน้น Skeleton + แนะนำ

**ต้องมีการเลือกอย่างน้อยโปรเจคละ 2 กลุ่ม

เกณฑ์การให้คะแนน (Scoring Guideline)

เต็ม 100 คะแนนต่อโปรเจกต์ — แบ่งตามองค์ประกอบ น้ำหนักคะแนนต่างกันตามความยากของแต่ละโจทย์

P1 = OCR Transcript (สจล.) · **P2 = OCR หลักสูตร + LLM ตอบคำถาม** · **P3 = Recommendation การต่อยมวย**

องค์ประกอบการให้คะแนน	P1	P2	P3	สิ่งที่ต้องทำได้ / เกณฑ์คะแนนเต็ม
1. ออกแบบฐานข้อมูล / RAG	15	20	10	Schema ชัดเจน ความสัมพันธ์ถูกต้อง รองรับการค้นคืน / query ได้จริง
2. การอ่านภาพ: OCR / Key point, Skeleton	35	20	25	ดึงข้อมูลจากภาพได้ครบฟิลด์, จุดสำคัญ ด้วย CV หรือ LLM-OCR / โมเดลดึง keypoint ที่ต้องใช้ได้ครบ/ การเลือกใช้ model (if ใช้ของพรี && ข้อความด้านบน = ได้เต็ม else มีการเสียเงิน ต้องพิจารณา เคสต่อว่า สมเหตุสมผลในเรื่องของเวลาประมวลผลและความแม่นยำไหม แพงไหม-เช่นอิงจากต่อคำถามหรือต่อผู้ใช้ 1 คน- เป็นต้น)
3. ความแม่นยำ / คุณภาพผลลัพธ์ (ทั้งภาพจำลองและของจริง)	40	20	35	วัดผลเป็นตัวเลขด้วย matrix ในเรื่องนั้นๆ : >91% = เต็ม, 80–90% = (~2/3), <80% = เริ่มต้น (~1/3)
4. LLM / Recommend	0	30	20	ตอบคำถาม ระดับง่าย-ยากได้ อ้างอิงข้อมูลจริง *ตอบถูก + อ้างอิงแหล่ง (หน้า/หัวข้อ ในเล่มหลักสูตร) จึงได้คะแนนเต็มของระดับนั้น* / ให้คำแนะนำได้ถูกต้อง ทั้งองศาของแต่ละส่วน และความเร็วที่ต้องปรับเปลี่ยน : >91% = เต็ม, 80–90% = ครึ่ง, <80% = เริ่มต้น
5. แอปใช้งานได้จริง + เอกสาร	10	10	10	Up code ขึ้น GitHub ที่แชร์กับ mail อ. (bhattarabhorn.wa at kmitl.ac.th) เป็น **collaborate** / Demo ทำงาน end-to-end พร้อม README / All Dataset / Presentation / Slide หรือ คู่มือการใช้งาน (ส่งใน discord กลุ่ม)
รวม	**100**	**100**	**100**	**คะแนนเต็มต่อโปรเจกต์ (น้ำหนักปรับตามความยากของโจทย์)**

ปล. Dataset สำหรับการ Challenge ส่วนใหญ่ต้อง ทำ label เอง

Challenge & คะแนนพิเศษ — P1: OCR Transcript

ของจริงไม่สะอาด: ไฟล์อัปโหลดมักมี noise และมีทรานสคริปต์ หลายเวอร์ชัน — ยิ่งทนทานและแม่นยำ ยิ่งได้คะแนนพิเศษ

เกณฑ์ Challenge	คะแนนพิเศษ	ระบบต้องทำได้ถึงจะได้คะแนน
ความแม่นยำ บนชุดภาพที่เป็นสถานการณ์จริง ของระดับปริญญาตรีและบัณฑิตศึกษา	+0	อ่านครบถูกแม้ภาพเอียง เงา ลายน้ำ หรือถ่ายจากมือถือ เครื่องสแกนหรือถ่ายเอกสาร
รองรับหลายเวอร์ชัน / ฟอร์แมต (≥ 2 แบบ)	+10	สลับฟอร์มเก่า/ใหม่ ของทรานสคริป สจล. โดยไม่ต้องแก้โค้ด
ความเร็ว (Latency): อ่าน 1 ฉบับพร้อมเปรียบเทียบความเหมือน เฉลี่ย < 30 วินาที	+10	ประมวลผลเร็ว รองรับงานจำนวนมาก (batch) ได้จริง

** นับคะแนนความแม่นยำสูงสุดเพียงระดับเดียว · รวมคะแนนพิเศษสูงสุด +10 คะแนน*

Challenge & คะแนนพิเศษ — P2: ถาม-ตอบหลักสูตร (LLM)

คะแนนพิเศษตาม “ระดับความยากของคำถาม” ที่ระบบตอบถูกและอ้างอิงหลักฐานในเล่มหลักสูตรได้

ตย. ระดับคำถาม	ตัวอย่างคำถาม	ทักษะที่ระบบต้องมี	คะแนนพิเศษ
ระดับ 1 ง่าย	“วิชาการเขียนโปรแกรมมีกี่หน่วยกิต?” / “รหัสวิชา 01006xxx ชื่ออะไร?”	ค้นหาฟิลด์/ข้อความตรง ๆ จากเล่มหลักสูตร + อ้างอิงหน้า/ข้อในเล่ม	+0
ระดับ 2 ปานกลาง	“วิชาบังคับชั้นปี 2 ภาคต้นมีอะไรบ้าง?” / “หมวดวิชาเฉพาะเลือกเก็บกี่หน่วยกิต?” / “วิชา X ต้องผ่านวิชาใดก่อน? ถ้ายังไม่ผ่านจะลงทะเบียนได้ไหม”	อ่านตาราง + รวม / กรองข้อมูลหลายจุด + อ้างอิงหน้า/ข้อในเล่ม	+0
ระดับ 3 ยาก	“ถ้าจะจบใน 3.5 ปี แต่ละเทอมต้องลงวิชาอะไร?” / “ตรวจว่าแผนเรียนนี้ครบเงื่อนไขจบหรือไม่” / “วิชาที่มีในหลักสูตรเก่า ไม่มีในหลักสูตรใหม่”	เชื่อมโยงเงื่อนไข prerequisite + ให้เหตุผล + อ้างอิงหน้า/ข้อในเล่ม	+0
ระดับ 4 ท้าทาย	สามารถ extract เล่มหลักสูตรหลาย version เช่น หลักสูตรของปีเดียวกัน แต่เป็นเวอร์ชันแก้ไข/ หลักสูตร 2 ปริญญา เป็นต้น	multi-hop + วางแผน + อ้างอิงหน้า/ข้อในเล่ม	+10
ความเร็ว (Latency)	ตอบ 1 คำถามในระดับยาก ภายใน < 5 วินาที (รวม retrieval + generate)	Index / RAG ที่ดึงเร็ว + ตอบไหลลื่น	+10

** รวมคะแนนพิเศษสูงสุด +10*

Challenge & คะแนนพิเศษ — P3: แนะนำการต่อยมวยไทย

คะแนนพิเศษ 2 แกน: (1) ยิ่งวิเคราะห์ท่ายากได้ และ (2) ยิ่งให้คำแนะนำที่นำไปแก้ได้จริง ยิ่งได้คะแนน

ระดับความสามารถ	สิ่งที่ระบบต้องทำได้ถึงจะได้คะแนน	คะแนนพิเศษ
แกนที่ 1 — ความยากของท่าที่วิเคราะห์ได้		
ท่าเดี่ยว (พื้นฐาน)	จับ keypoint หมัด / เตะพื้นฐานได้ถูกต้อง	+0
ท่าผสมหลายส่วนพร้อมกัน	แยกแยะ หมัด+เตะ / ศอก+เข่า / การ์ด+สวนกลับ ในจังหวะเดียว	+5
จังหวะต่อเนื่อง (คอมโบ)	วิเคราะห์ลำดับท่า + timing + การทรงตัวขณะเปลี่ยนท่า	+10
แกนที่ 2 — คุณภาพคำแนะนำ (Recommendation)		
ชี้จุดที่ต้องแก้	เทียบกับครูฝึก มีทั้ง text และ เป็นค่าตัวเลข เช่น มุมข้อศอก ความสูงการเตะ ตำแหน่งการ์ด	+0
แนะนำที่ทำตามได้จริง	บอกวิธีปรับ + จัดลำดับความสำคัญ + วัดพัฒนาการเมื่อฝึกซ้ำ	+5
ความเร็ว / เรียลไทม์ (Real-time)	ให้ feedback ภายใน < 5 วินาที	+10

- *คะแนนพิเศษรวมสูงสุด +10 · โดยการแนะนำหรือวิเคราะห์ท่าต้องเทียบกับท่ามาตรฐานของครูมวย*

15-WEEK COURSE SCHEDULE · Intelligent System Development · Undergraduate · 3+ hrs/session · Lecture + Lab

Category Key: Foundation, Core AI/ML, LLMs & APIs, Engineering, Capstone

Wk	Topic	Theory / Lecture	Lab Activity	Project Milestone	Checkpoint
1	Intro + Git Setup	AI/ML/DL hierarchy · AI stack · Git	Env setup · API call · git init	จัดกลุ่ม + ตั้ง repo เลือก domain	—29/6
2	Computer Vision & AI	Vision basics · Classical CV · CNN	Augmentation and labeling Dataset project (ของอ.)	--	--6/7
3	ML & DL Foundations CNN	Sup/Unsup/RL · Gradient · MLP	Dataset Lab 2 มา Run model	--	--13/7
4	How to build CNN/ RNN	How to build CNN/ RNN	Augmentation and labeling Dataset project (challenge)	ยืนยัน Dataset	✅ Checkpoint 1 20/7
5	Generative Modeling	Autoencoder · VAE · GAN · Diff	Dataset Lab 4 มา Run model	แยกกัน Run model	— 27/7
6	Deep Reinforcement Learning	RL · Q-learning · DQN · policy	คิดคำถาม / คำแนะนำ / ค่าที่จะ evaluate ทั้งหมด	ส่ง Proposal	📄 Proposal Due 3/8
7	Agentic AI & LLMs	Agentic AI & LLMs	LLMOCR extract document / Algorithm เปรียบเทียบท่วงท่า	เพิ่ม LLM เข้า project	—10/8
			Midterm เนื้อหา Wk 1-6 (17/8 เช้า ต้องดูตารางสอบอีกทีนะ)		
8	Transfer & Fine-tune	pre-trained · fine-tune	LLMOCR extract document / Algorithm เปรียบเทียบท่วงท่า		—24/8
			หยุด 31/8 เตรียมลาดกระบังนิทรรศ + open house		
9	Evaluation & Overfit	bias-variance · metrics · reg	Run model ที่คาดว่าจะใช้ใน project ให้ครบ loop	ประเมิน metrics ทุก model	✅ Checkpoint 2 7/9
10	AI APIs & Integration	API · embeddings · function call	ทำโมเดลให้เก่งขึ้น และเลือกโมเดลในกลุ่ม	Model ทำงานได้ ครบ loop	🔌 Integration spike 14/9
11	Frontend Development	HTML/CSS · JS · responsive	Index.html	wireframe + เชื่อม UI	—21/9
12	Backend Development	REST · FastAPI · auth · DB	ทั้งระบบเชื่อมต่อกัน	End to end	✅ Checkpoint 3 28/9
13	สรุปภาพรวม Project	สรุป Project + ทดสอบ Challenge	สรุป Project + ทดสอบ Challenge	--	—ทดสอบ challenge 5/10
14	ทำเอกสาร + ทำ presentation	ทำเอกสาร + presentation	ทำเอกสาร + presentation (อันนี้เดี๋ยว vote : online/onsite)	🔒 Final code freeze	🚫 Code Freeze 12/10
15	Final Presentations	present + demo · review · Q&A	Live demo full-stack AI	ส่ง Code + Slides + Demo	Presentation Day 19/10

Team Project Milestones & Grading			
Milestone	สัปดาห์	รายละเอียด	Deliverable
จัดกลุ่ม + เลือก Domain	1	กลุ่มละ 3-4 คน · ตกลง domain	ชื่อกลุ่ม + domain ส่ง LMS
Checkpoint 1	4	ยืนยัน รูปแบบ dataset ที่จะใช้ในโปรเจคว่า มีเคสจำลองแบบไหนบ้าง	Slide 2-3 หน้า
Project Proposal	6	อธิบาย problem · AI approach · tech stack · timeline	PDF 1 หน้า (A4)
Checkpoint 2	8	Model train ได้ · มี evaluation metrics ·ทุกโมเดล และสรุปโมเดลที่เลือกพร้อมเหตุผล	Code.py + metrics table
Mid-project Review	10	Instructor feedback session · path model เชื่อมต่อกันได้อย่างชัดเจน	Demo (5 นาที) + Q&A
Checkpoint 3	12	Frontend + AI + Backend integrated ครบ · ทดสอบ end-to-end	Working prototype (live demo)
Code Freeze	14	หยุด add feature ใหม่ · polish + bugfix + Docker deploy	GitHub repo link (final commit)
Final Presentation	15	Demo 10 นาที + Q&A 3 นาที · peer review form	Slides + Live Demo
Written Report	15	Architecture · results · challenges · reflection (max 10 หน้า)	Dataset + PDF report + GitHub link

ภาพรวมรายวิชา — วิชานี้เรียนอะไร ทำอะไรได้

PHASE 1

Foundations

สัปดาห์ 1–4

- AI / ML / DL พื้นฐาน
- Computer Vision
- CNN · RNN · Attention
- พื้นฐานการสร้างโมเดล

PHASE 2

Modeling

สัปดาห์ 5–7

- Generative Models
- Reinforcement Learning
- Fine-tuning โมเดล

PHASE 3

LLM & Integration

สัปดาห์ 8–10

- LLM & Prompting
- เรียก AI ผ่าน API
- Frontend เบื้องต้น

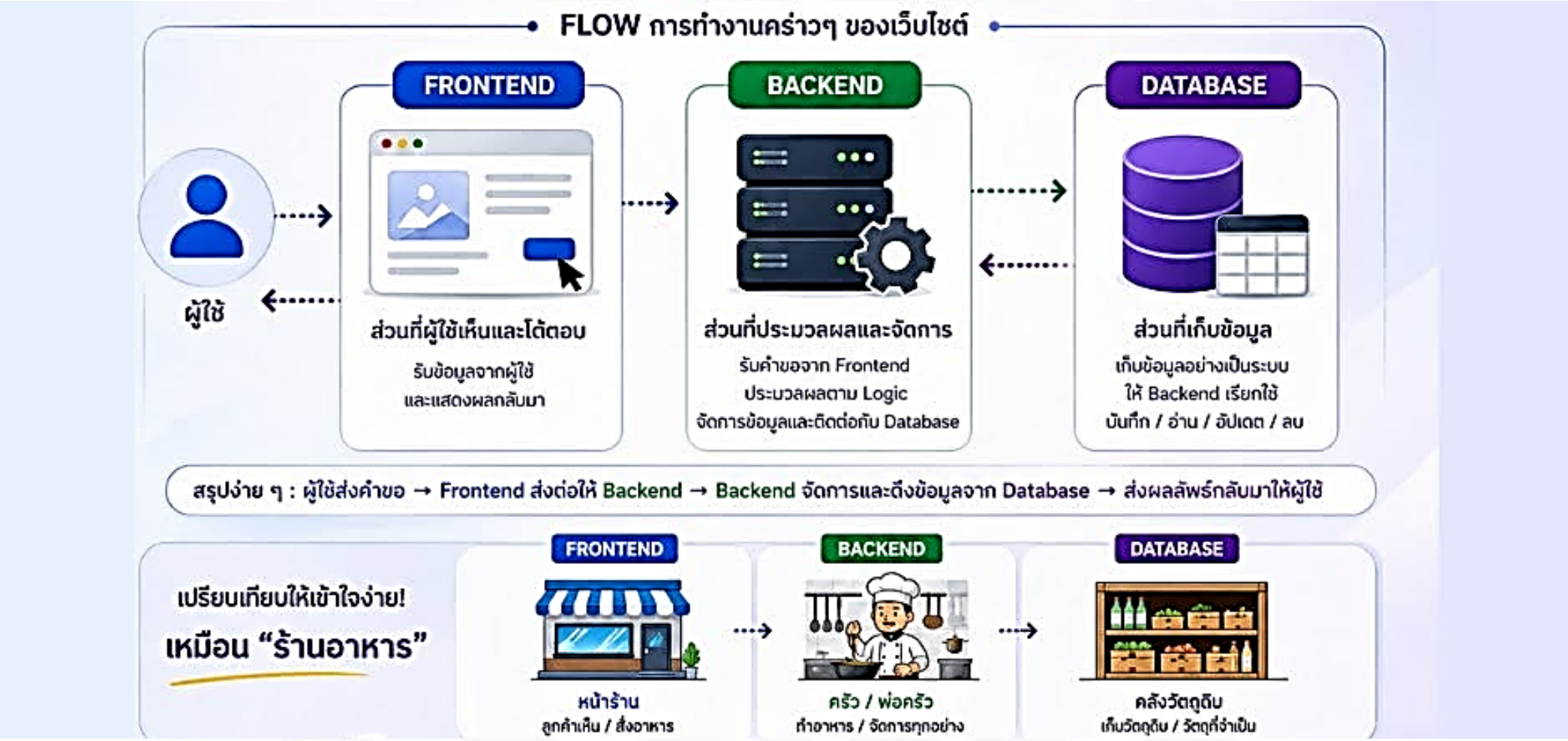
PHASE 4

Full-stack Product

สัปดาห์ 11–15

- Frontend
- Backend
- Deploy ด้วย Docker
- นำเสนอโปรเจกต์

เชื่อมทุกส่วนด้วยโปรเจกต์เดียว — ใช้ Git จัดการโค้ดตั้งแต่สัปดาห์แรก จนถึงงานนำเสนอสุดท้าย


FLOW การทำงานคร่าวๆ ของเว็บไซต์
FRONTEND
BACKEND
DATABASE
ผู้ใช้
ส่วนที่ผู้ใช้เห็นและโต้ตอบ
รับข้อมูลจากผู้ใช้
และแสดงผลกลับมา
ส่วนที่ประมวลผลและจัดการ
รับคำขอจาก Frontend
ประมวลผลตาม Logic
จัดการข้อมูลและติดต่อกับ Database
ส่วนที่เก็บข้อมูล
เก็บข้อมูลอย่างเป็นระบบ
ให้ Backend เรียกใช้
บันทึก / อ่าน / อัปเดต / ลบ
สรุปง่าย ๆ : ผู้ใช้ส่งคำขอ → Frontend ส่งต่อให้ Backend → Backend จัดการและดึงข้อมูลจาก Database → ส่งผลลัพธ์กลับมาให้ผู้ใช้
เปรียบเทียบให้เข้าใจง่าย!
เหมือน "ร้านอาหาร"
FRONTEND
BACKEND
DATABASE
หน้าร้าน
ลูกค้าเห็น / สั่งอาหาร
ครัว / พ่อครัว
ทำอาหาร / จัดการทุกอย่าง
คลังวัตถุดิบ
เก็บวัตถุดิบ / วัตถุที่จำเป็น

AI / ML / DL — ลำดับชั้นของปัญญาประดิษฐ์

วงในเป็นส่วนหนึ่งของวงนอก — DL ⊂ ML ⊂ AI

AI — Artificial Intelligence

ทำให้เครื่องทำงานที่ต้องใช้ความฉลาดแบบมนุษย์ (รวมถึงกฎ if-else แบบดั้งเดิม)

เช่น ระบบแนะนำสินค้า · ผู้ช่วยเสมือน

ML — Machine Learning

ให้เครื่องเรียนรู้รูปแบบจากข้อมูลเอง โดยไม่ต้องเขียนกฎทุกกรณี

เช่น คัดกรองสแปม · พยากรณ์ยอดขาย

DL — Deep Learning

ML ที่ใช้โครงข่ายประสาทเทียมหลายชั้น เหมาะกับข้อมูลซับซ้อน

เช่น รู้จำภาพ · แปลภาษา · เป็นพื้นฐานของ LLM

The AI Stack — มอง AI เป็นชั้น ๆ

งาน AI แบ่งเป็นชั้น ๆ — ชั้นล่างเป็นพื้นฐาน ชั้นบนใกล้ผู้ใช้ · คอลัมน์ขวาบอกว่ารายวิชาลงลึกแต่ละชั้นสัปดาห์ใด

ชั้นสูงขึ้น = ใกล้ผู้ใช้มากขึ้น

		เรียนสัปดาห์
Application / UI	เว็บแอป · แชตบอต · แดชบอร์ด	W11–14
API / Serving	REST · FastAPI · Model API	W9–10, 13
Models	CNN · Transformer · LLM	W3–8
Frameworks / Libraries	PyTorch · TensorFlow · scikit-learn	W2–7
Compute	GPU · Cloud · Colab	ตลอดเทอม
Data	Datasets · Storage · Labeling	W2–5

ภาพรวมระบบ: Frontend → API → Backend → AI Model

ผู้ใช้คลิกบนหน้าเว็บ → คำขอวิ่งผ่าน API → Backend ประมวลผลและเรียก AI Model → ส่งผลลัพธ์กลับมาแสดงผล

Frontend
หน้าเว็บที่ผู้ใช้เห็นและคลิก
`.html` `.css` `.js`

→

API
ตัวกลาง: รับ request แล้วส่งต่อ
`routes · /api/...`

→

Backend
ตรรกะ/ประมวลผลของระบบ
`app.py · .py`

→

AI Model
ทำนาย/วิเคราะห์ข้อมูล
`model.pkl` `.h5` `.onnx`

↕

Database (ฐานข้อมูล)
`.db` `.sqlite` `.csv`

อ่านลูกศรอย่างไร

→ request ขาไป · ↕ อ่าน/เขียนฐานข้อมูล

Response (ขากลับ): ผลลัพธ์ JSON ถูกส่งจาก Backend → API → Frontend เพื่อแสดงผลให้ผู้ใช้

โครงสร้างโฟลเดอร์มาตรฐาน & ชนิดไฟล์ในแต่ละส่วน

โปรเจกต์มาตรฐานจัดไฟล์เป็นชั้นๆ — แยก frontend / backend / model / data ออกจากกันชัดเจน

project structure

```
intelligent-system/
├─ frontend/
│   ├─ index.html
│   ├─ style.css
│   └─ app.js
├─ backend/
│   ├─ app.py          ← Flask / FastAPI
│   ├─ api/            ← จุดที่ API อยู่
│   │   └─ routes.py
│   └─ requirements.txt
├─ model/
│   ├─ train.py
│   └─ model.pkl       ← โมเดล AI
├─ data/
│   └─ dataset.csv
├─ .gitignore
└─ README.md
```

frontend/ `.html` `.css` `.js`
หน้าเว็บ — สิ่งที่ผู้ใช้เห็นและคลิก

backend/ `.py` `(app.py, routes.py)`
ตรรกะระบบ + จุดที่ API ทำงาน

model/ `.pkl` `.h5` `.onnx`
โมเดล AI ที่เทรน/บันทึกไว้แล้ว

data/ `.csv` `.json`
ชุดข้อมูลสำหรับเทรน/ทดสอบ

ราก (root) `README.md` · `requirements.txt` · `.gitignore`
เอกสาร · รายการไลบรารี · ไฟล์ที่ไม่ push

Git — ระบบควบคุมเวอร์ชันของโค้ด

Git = ระบบควบคุมเวอร์ชัน (Version Control) ที่บันทึกประวัติการเปลี่ยนแปลงโค้ด — ย้อนกลับได้ และทำงานเป็นทีมได้

ย้อนเวลาได้	ทำงานเป็นทีม	ติดตามการแก้ไข	สำรองบนคลาวด์
กลับไปเวอร์ชันก่อนหน้าได้เมื่อโค้ดพัง	หลายคนแก้ไฟล์เดียวกัน แล้ว merge รวมงาน	รู้ว่าใครแก้อะไร เมื่อไหร่ ทุกครั้ง	push ขึ้น GitHub ไม่กลัวเครื่องพัง

Git Workflow — เส้นทางของโค้ดจากเครื่องเราขึ้น GitHub

Working Directory — โฟลเดอร์ที่เราแก้โค้ด
→ `git add`
Staging Area — เลือกไฟล์ที่จะบันทึก
→ `git commit`
Local Repo — บันทึกในเครื่อง (commit)
→ `git push`
Remote · GitHub — สำรองบนคลาวด์

ทำงานที่ไหน? เครื่อง Local ⟷ GitHub (คลาวด์)

โค้ดอยู่ 2 ที่: เครื่องของคุณ (Local) กับ GitHub (คลาวด์) — แก้ที่ Local แล้ว “sync” ขึ้น/ลงกับ GitHub

เครื่อง Local (คอมของคุณ)

- ที่ที่คุณ **เขียนและรันโค้ดจริง**
- โฟลเดอร์โปรเจกต์เก็บอยู่ในเครื่อง แก้แล้วเห็นผลทันที
- ต้องติดตั้ง Git และ Python ไว้ในเครื่อง
- งานยัง **ไม่ถูกสำรอง** จนกว่าจะ push ขึ้นไป

push → อัปขึ้น GitHub

pull ← ดึงลงเครื่อง

GitHub (บนคลาวด์)

- เก็บโค้ด **ส่วนกลาง + สำรองออนไลน์**
- เพื่อนในกลุ่ม ดึงไปทำต่อ ได้พร้อมกัน
- เก็บ **ประวัติการแก้ทุกเวอร์ชัน**
- ใช้ส่งงาน / แชร์ลิงก์ เข้าผ่านเว็บ github.com

กฎง่ายๆ: แก้ที่ Local เสมอ ⟶ commit ⟶ push ขึ้น GitHub · เริ่มงานใหม่ทุกครั้งให้ pull ก่อน

เริ่มต้น: นำโปรเจกต์ลงเครื่อง (clone) — 3 ทาง

เป้าหมายเดียวกันทั้ง 3 วิธี: ดึง repo จาก GitHub มาไว้ในเครื่อง (ทำครั้งเดียวตอนเริ่มโปรเจกต์)

GitHub Desktop

แอปในเครื่อง · ง่ายสุด

1. เปิดแอป แล้วไปที่ File ▶ Clone repository
2. เลือก repo ของกลุ่ม และโฟลเดอร์ปลายทาง
3. กด Clone ⟶ ได้โปรเจกต์ในเครื่องทันที

ไม่ต้องพิมพ์คำสั่งเลย เหมาะกับมือใหม่

เว็บ GitHub (UI)

บนเบราว์เซอร์ · github.com

1. เข้า github.com แล้วเปิด repo ของกลุ่ม
2. กดปุ่ม Code (เขียว) ⟶ คัดลอกลิงก์ HTTPS
3. นำลิงก์ไป clone ด้วย Desktop หรือ cmd

เว็บใช้สร้าง repo / อัปไฟล์ทีละไฟล์ / ดูประวัติ

Command line (cmd)

เทอร์มินัล · ยืดหยุ่นสุด

1. เปิด cmd ในโฟลเดอร์ที่จะเก็บงาน

```
cd Documents
```

2. ดึง repo ลงมา (clone)

```
git clone …/isd-2025-group01.git
```

3. เข้าไปในโฟลเดอร์โปรเจกต์

```
cd isd-2025-group01
```

<ลิงก์> ได้จากปุ่ม Code สีเขียวบนเว็บ

วงจรทำงานประจำวัน: pull → แก้ → commit → push

ทุกครั้งที่จะทำงาน: ดึงงานล่าสุดก่อน (pull) → แก้โค้ด → บันทึกเป็นเวอร์ชัน (commit) → ส่งขึ้น GitHub (push)

1 Pull
ดึงงานล่าสุด

→

2 แก้โค้ด
เขียน · แก้ไฟล์

→

3 Commit
บันทึกเวอร์ชัน

→

4 Push
ส่งขึ้น GitHub

↺ เสร็จแล้ววนกลับไป pull ใหม่ทุกครั้งที่จะเริ่มงานรอบถัดไป

GitHub Desktop

commit + push แบบกดปุ่ม

1 ดูไฟล์ที่แก้ในช่อง Changes (ซ้ายมือ)

2 พิมพ์ข้อความสรุปใน Summary แล้วกด Commit to main

3 กดปุ่ม Push origin (มุมขวาบน) → ขึ้น GitHub

เว็บ GitHub (UI)

เหมาะแก้ไฟล์ย่อยๆ บนเบราว์เซอร์

1 เปิดไฟล์บนเว็บ แล้วกดไอคอนดินสอ (Edit)

2 แก้ข้อความ แล้วเลื่อนลงด้านล่าง

3 กด Commit changes → เว็บ commit + push ให้เลย

Command line (cmd)

พิมพ์ 3 คำสั่งตามลำดับ

1 เพิ่มไฟล์ที่แก้ทั้งหมด

```
git add .
```

2 บันทึกเวอร์ชัน + ใส่ข้อความ

```
git commit -m "ข้อความ"
```

3 ส่งขึ้น GitHub

```
git push
```

Lab 1

Frontend	API	Backend	AI Model	Database

Lab 1

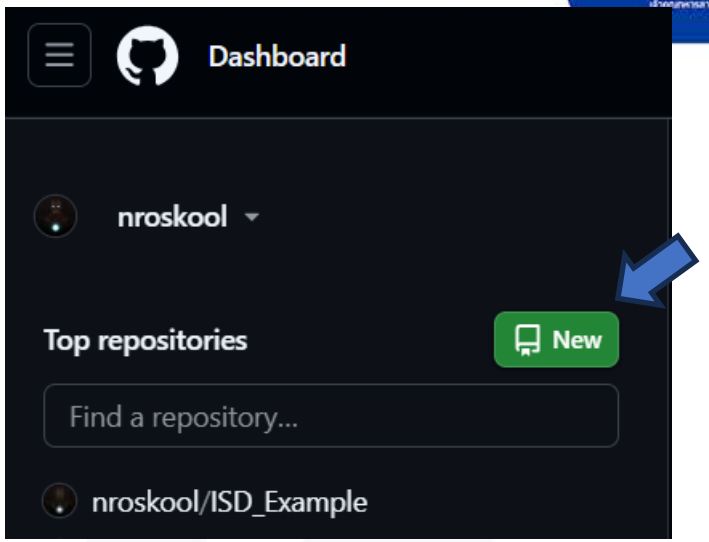
สร้าง Git Repo แรกของกลุ่ม

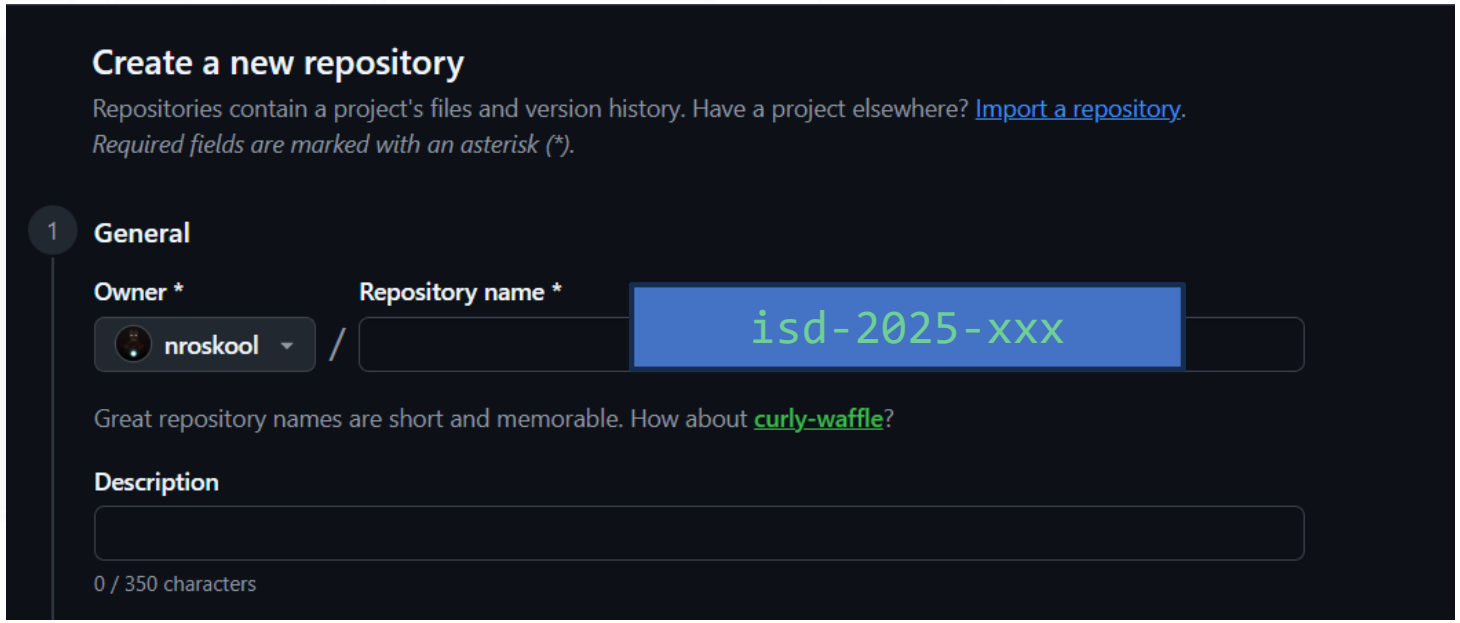
ส่วนของระบบ: การทำงานร่วมกันบน GitHub (รอบ pull / commit / push)

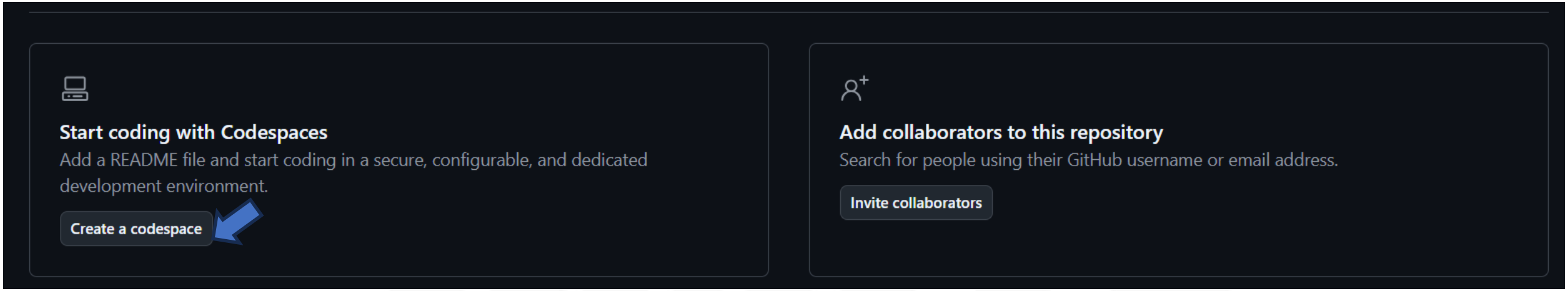
นำโค้ดทั้งโปรเจกต์ขึ้น GitHub เพื่อเก็บ สำรอง และทำงานเป็นทีม

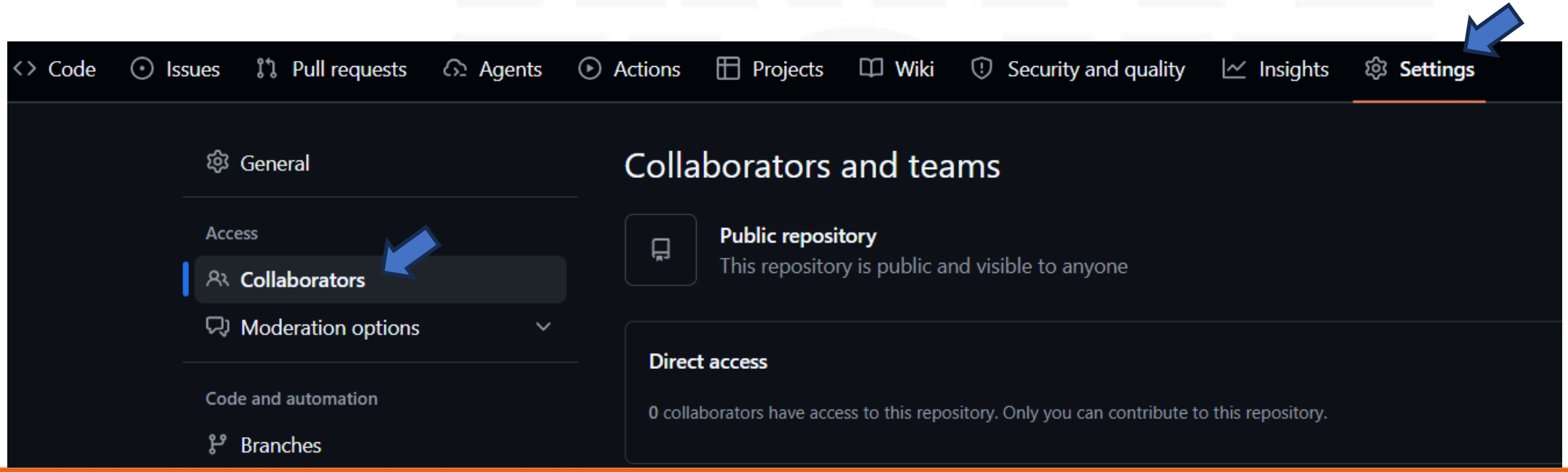
การสร้าง GitHub rerpository บน web

- https://github.com/ >> login >> ตัวแทนกลุ่มสร้าง Repository






Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
Create a codespace
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators


Code
Issues
Pull requests
Agents
Actions
Projects
Wiki
Security and quality
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Collaborators and teams
Public repository
This repository is public and visible to anyone
Direct access
0 collaborators have access to this repository. Only you can contribute to this repository.

ISD_Example Private

Watch 0

Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. View documentation.

Dismiss

Protect this branch

main | 2 Branches | 0 Tags | Go to file T | Add file | Code

nroskool first 6daa372 · 19 hours ago 1 Commit

hello.py first 19 hours ago

README

Go to file T | Add file | Code

Local | Codespaces

Clone

HTTPS | SSH | GitHub CLI

https://github.com/nroskool/ISD_Example.git

Clone using the web URL.

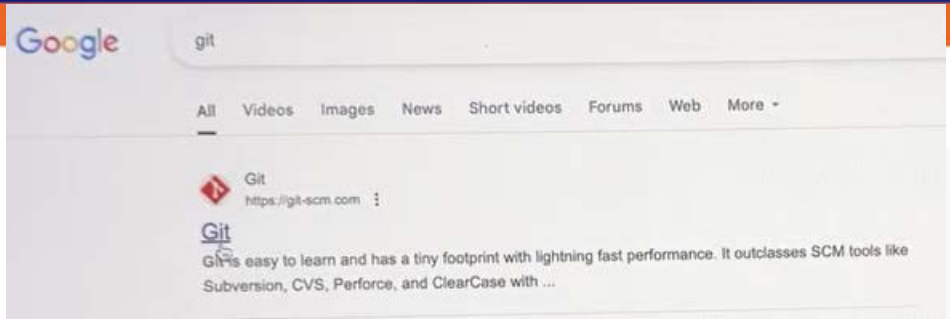
Open with GitHub Desktop

Download ZIP

a READM

an overview

ติดตั้ง Git



- เปิด terminal >> ไป path ที่ต้องการสร้างโฟลเดอร์งาน

```
C:\Users\NROSKOOL>D:

D:\>cd D:\GitHub

D:\GitHub>
```

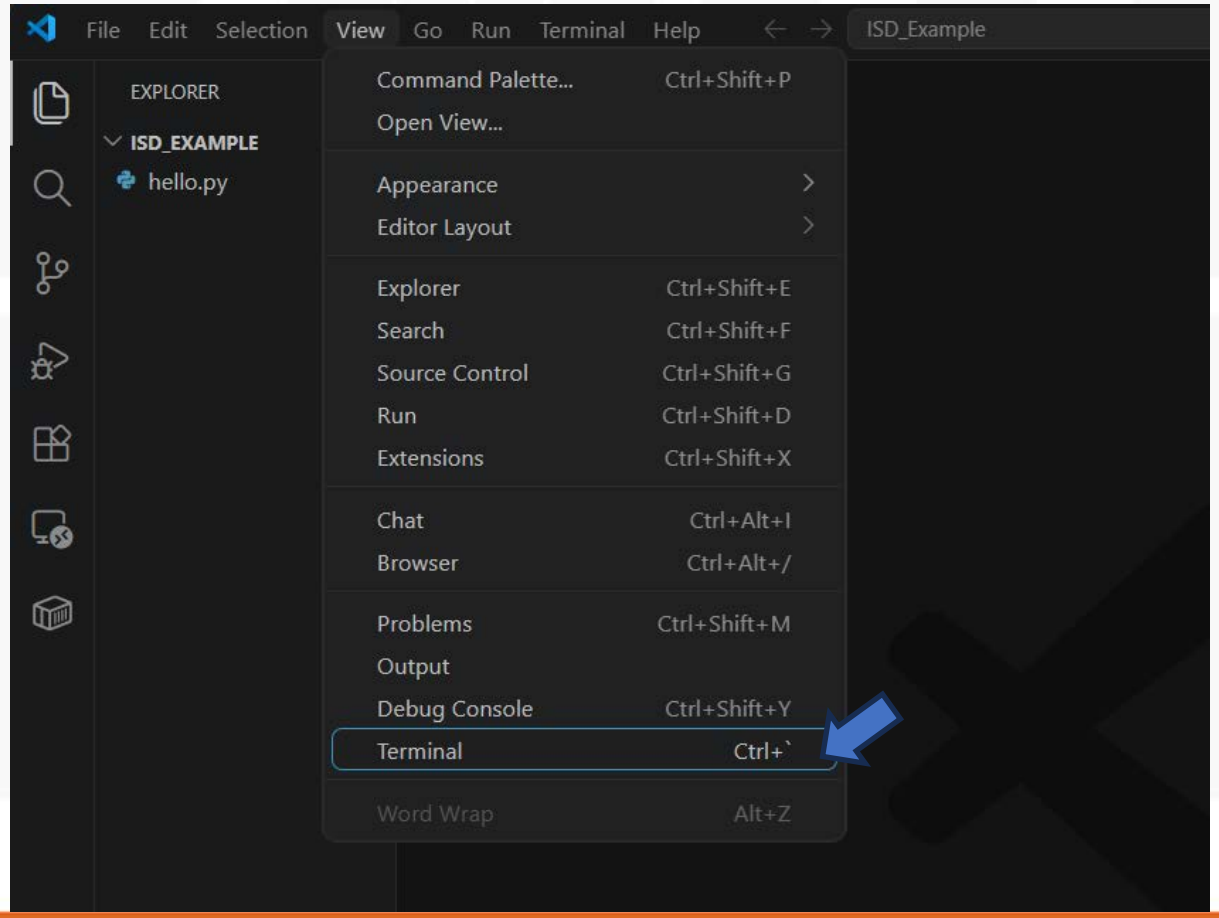
- จากนั้นใช้คำสั่ง **git clone** ตามด้วย url ที่ก๊อบมาจากสไลด์ก่อนหน้า
- เมื่อ clone มาเสร็จ cd เข้าไปในโฟลเดอร์ที่โคลนลงมาด้วยคำสั่ง **cd ISD_Exxample**
- จากนั้นพิมพ์ **code .** เพื่อเข้าสู่ vscode

```
D:\GitHub>git clone https://github.com/nroskool/ISD_Example.git
Cloning into 'ISD_Example'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), done.

D:\GitHub>cd ISD_Example

D:\GitHub\ISD_Example>code .
```

Feature branch

ห้ามทำงานบน branch main ให้ทำบน branch ตัวเอง ดังนั้น

เริ่มแรก update local repo (ใน vscode ในเครื่องเรา) ให้มี code อะไรต่างๆเท่า บน git ซึ่งคือ remote repo ก่อน



Feature branch

- **git checkout main** เพื่อดูว่าเราอยู่ที่ main หรือป่าว ถ้าใช่จะขึ้น Already on 'main'
- **git pull origin main** เพื่อ pull ทุกอย่างที่ main มีลงมา

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\GitHub\ISD_Example> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS D:\GitHub\ISD_Example> git pull origin
Already up to date.
PS D:\GitHub\ISD_Example> git pull origin main
From https://github.com/nroskool/ISD_Example
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\GitHub\ISD_Example>
```

Feature branch

สร้าง branch ที่จะทำงาน ด้วยคำสั่ง **git checkout -b feature/index** เพื่อเติม code ต่างๆที่ต้องการบน branch ที่สร้างมาใหม่นั้น

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\GitHub\ISD_Example> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS D:\GitHub\ISD_Example> git pull origin
Already up to date.
PS D:\GitHub\ISD_Example> git pull origin main
From https://github.com/nroskool/ISD_Example
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\GitHub\ISD_Example> git checkout -b feature/index
```

OUTLINE
TIMELINE
main 0 0 Git Graph

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\GitHub\ISD_Example> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS D:\GitHub\ISD_Example> git pull origin
Already up to date.
PS D:\GitHub\ISD_Example> git pull origin main
From https://github.com/nroskool/ISD_Example
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\GitHub\ISD_Example> git checkout -b feature/index
Switched to a new branch 'feature/index'
PS D:\GitHub\ISD_Example>
```

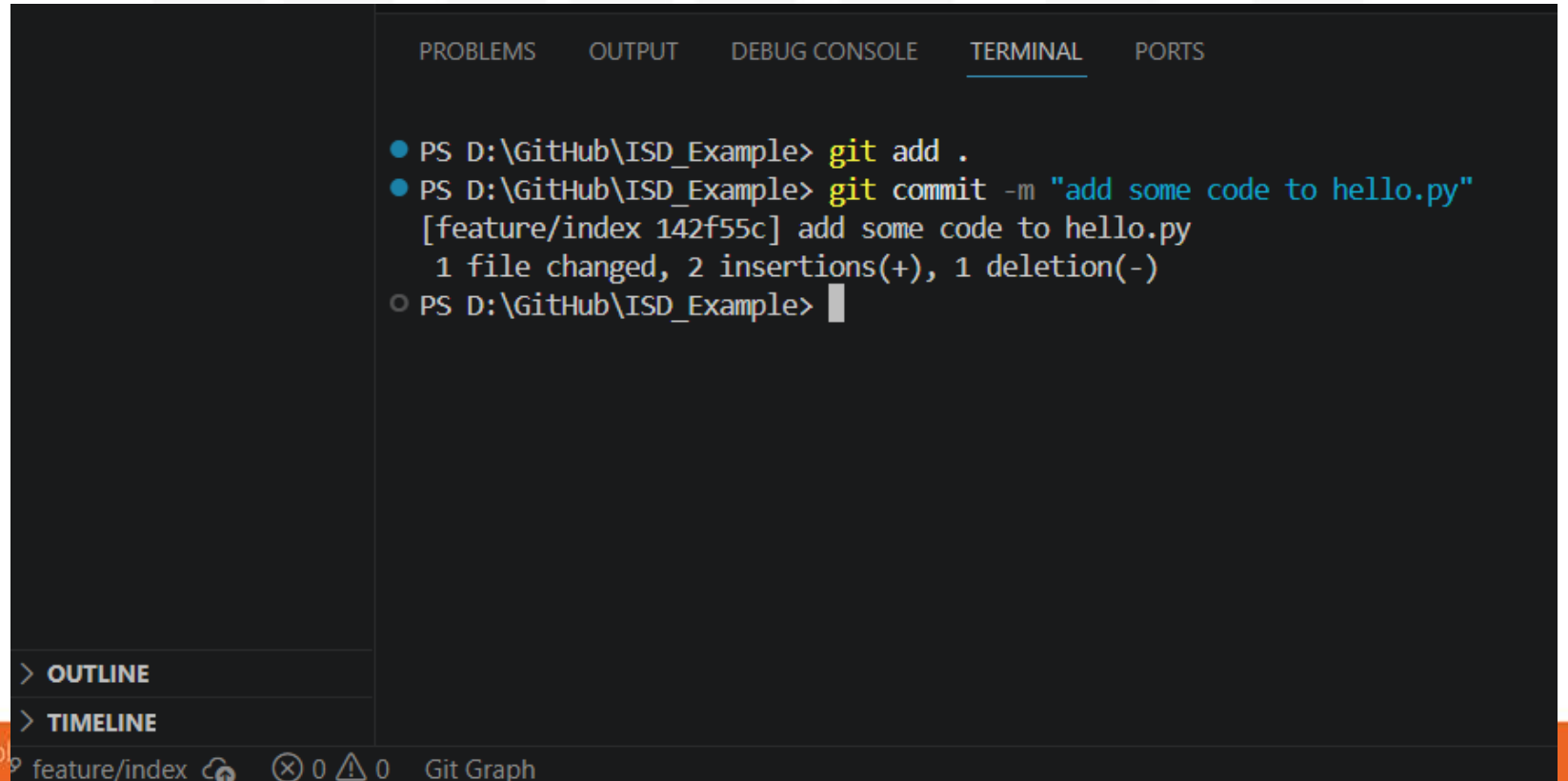
OUTLINE
TIMELINE
feature/index 0 0 Git Graph

Feature branch

เมื่อเพิ่มหรือแก้จนพอใจ จะ push code ขึ้น branch ตัวเอง ใช้คำสั่ง

git add ตามด้วยชื่อไฟล์.นามสกุลไฟล์ หรือ ถ้ามีหลายไฟล์ อยาก push พร้อมกันหมด ให้ใช้คำสั่ง **git add .**

จากนั้น add commit เพื่อ ใส่ข้อความใน “ ” ว่า commit นี้เราทำอะไรไป ด้วย **git commit –m “ข้อความ”**



Feature branch

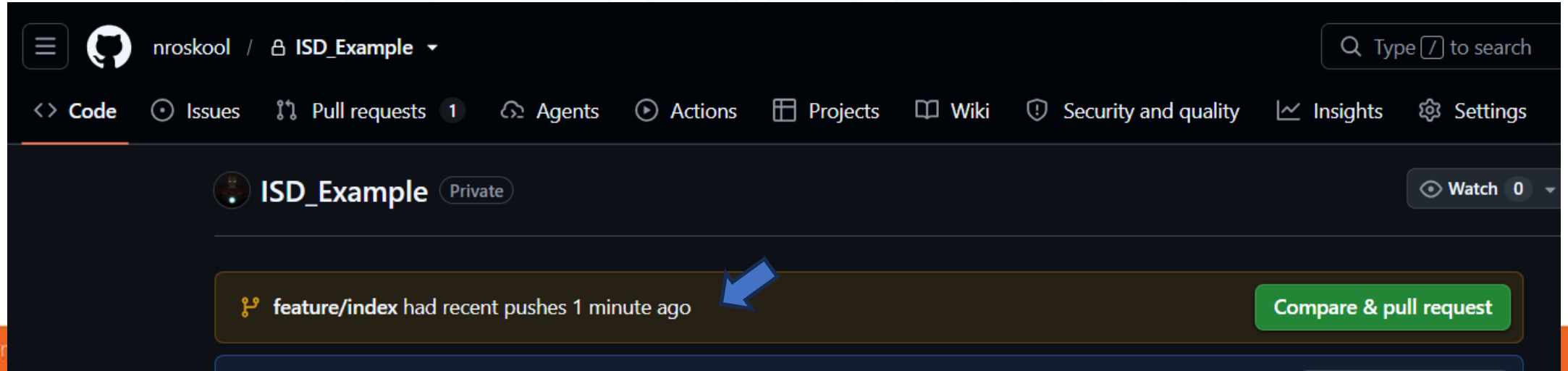
จากนั้นเราจะ push ขึ้น repository เพื่อให้คนอื่นเห็นงานเรา และเป็นการสำรองโค้ดไว้บน rermote

ด้วยคำสั่ง **git push –u origin ตามด้วยชื่อbranch**

```
PS D:\GitHub\ISD_Example> git push -u origin feature/index
branch 'feature/index' set up to track 'origin/feature/index'.
Everything up-to-date
PS D:\GitHub\ISD_Example>
```

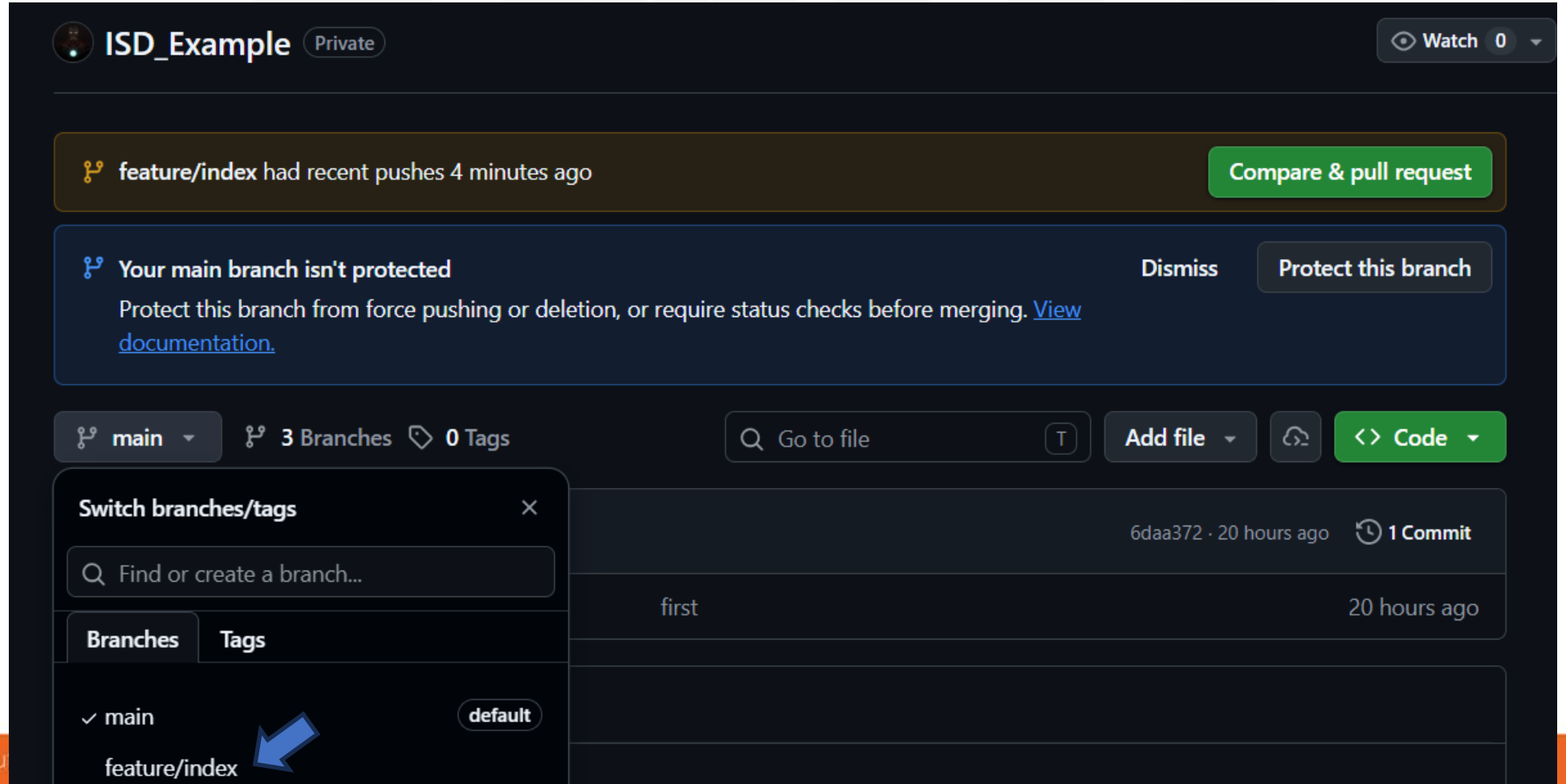
OUTLINE
TIMELINE
feature/index 0 0 Git Graph

สามารถมาดูที่เว็บ GitHub ได้ เพพื่อความมั่นใจว่า push สำเร็จ จะมีตรงนี้ขึ้นมา



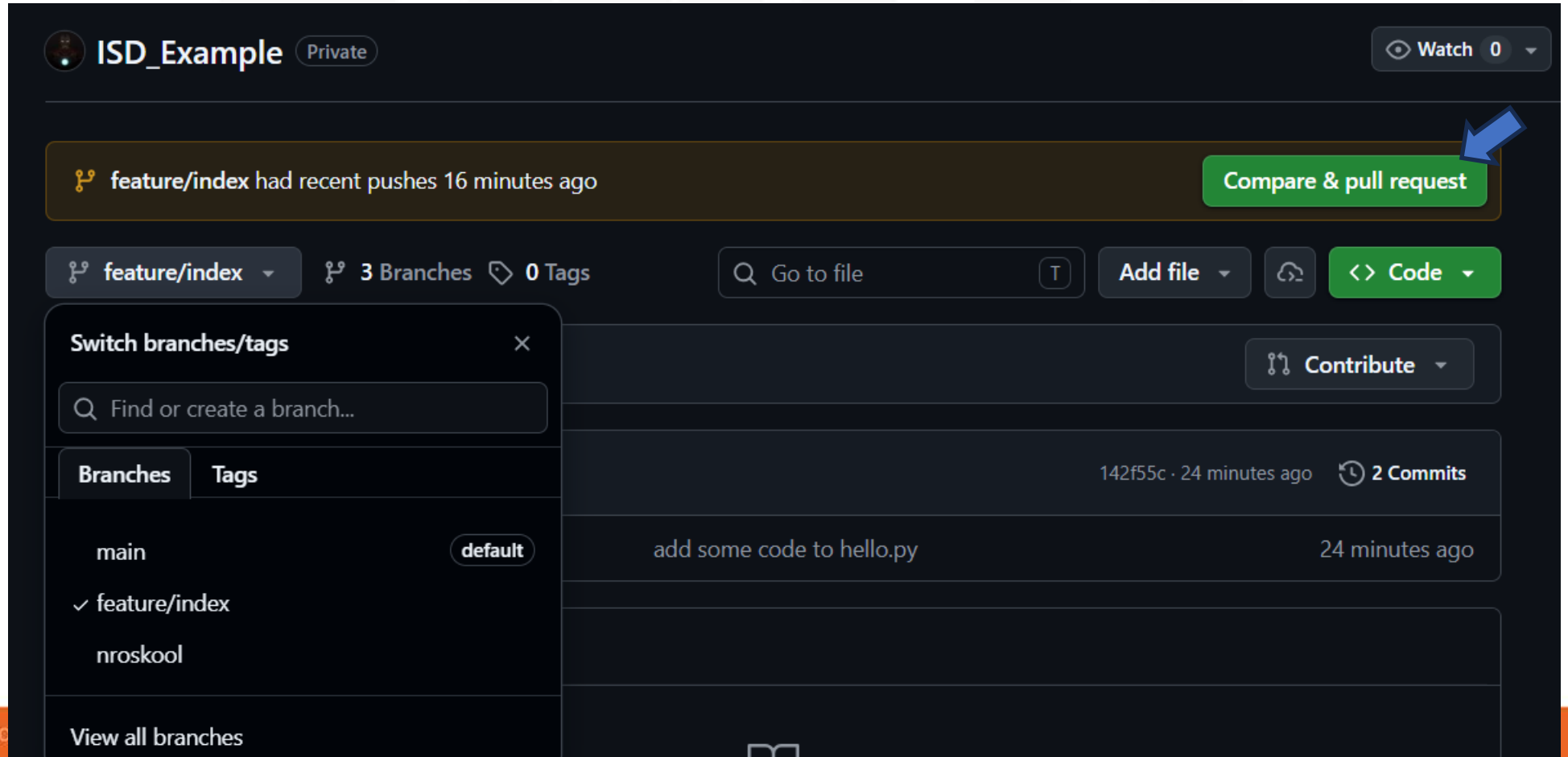
Feature branch

- Refresh แล้วดูจะมี branch เพิ่มขึ้นมา

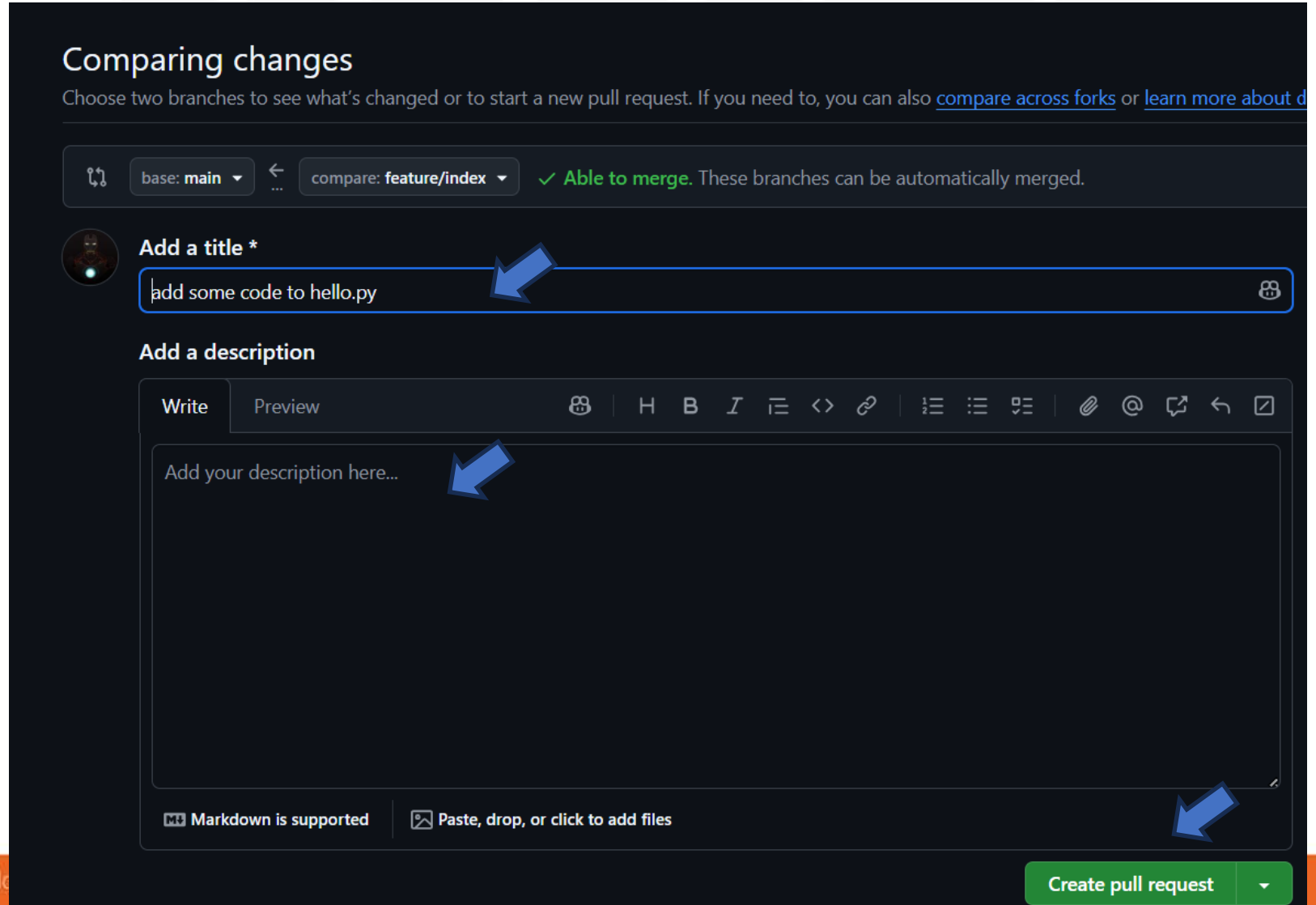


Feature branch

ขั้นต่อมา Merge branch เพื่อ นำ branch ที่เพิ่มทำงาน เข้า branch กลาง



Feature branch



Feature branch

add some code to hello.py #2

Open nroskool wants to merge 1 commit into main from feature/index

Conversation 0 | Commits 1 | Checks 0 | Files changed 1

nroskool commented now

No description provided.

142f55c

View session

add some code to hello.py #2

Open nroskool wants to merge 1 commit into main from feature/index

Conversation 1 | Commits 1 | Checks 0 | Files changed 1

All commits

Filter files...

hello.py

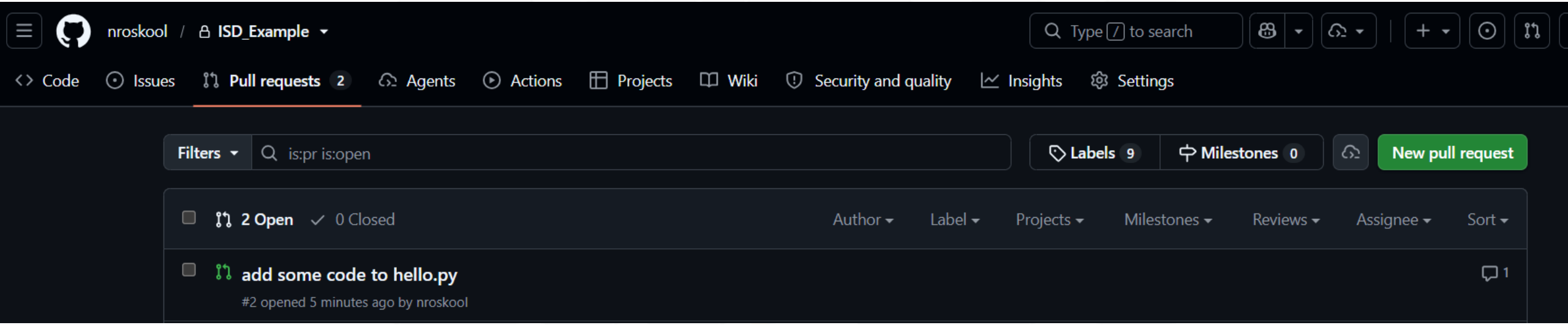
hello.py

```
@@ -1 +1,2 @@
1  - print("Hi")
1  + print("Hi")
2  + print("Hello world")
```

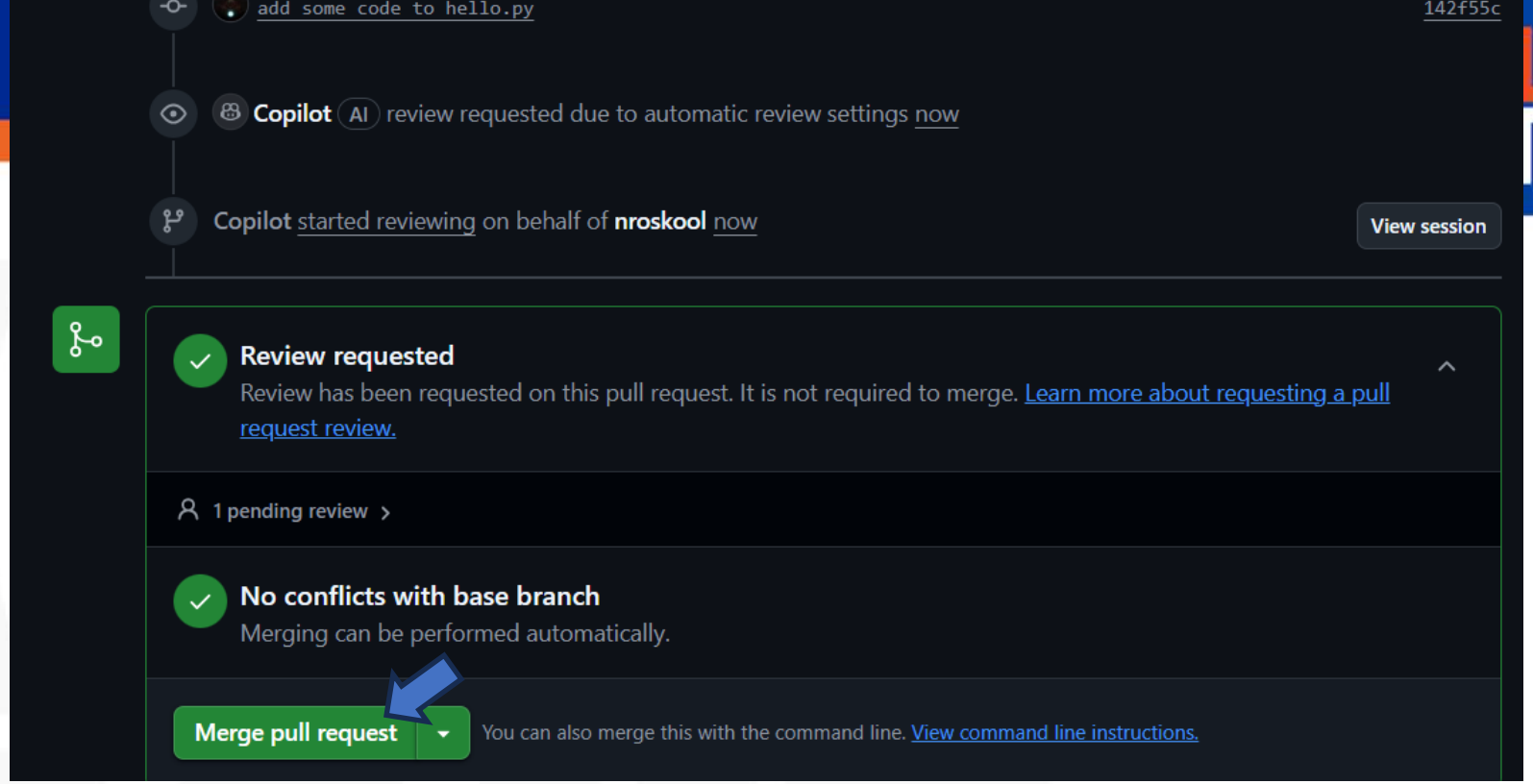
ore about requesting a pull

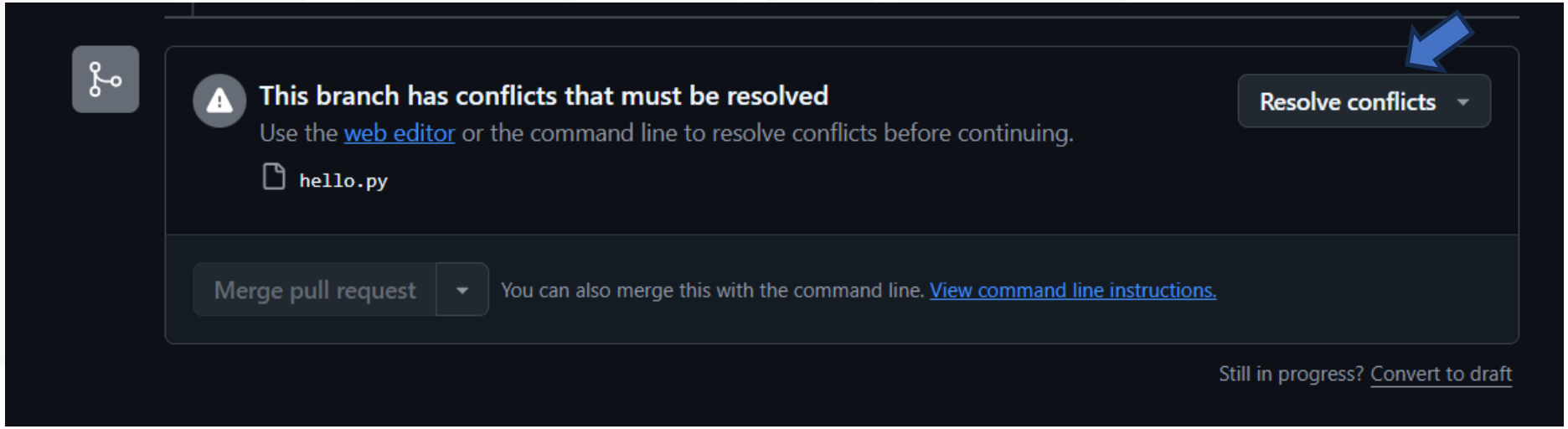
structions.

Feature branch

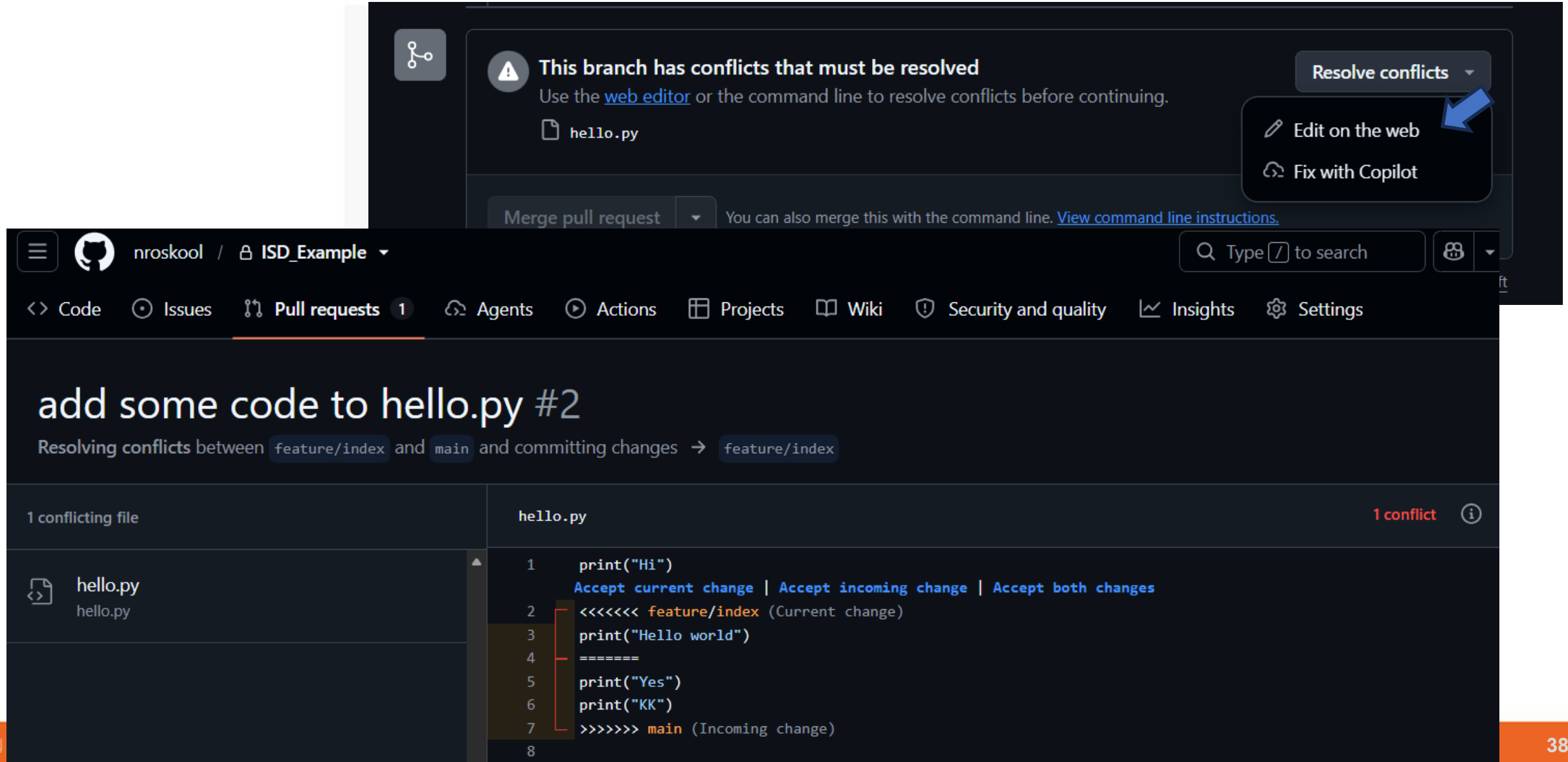


Feature branch





Feature branch



Feature branch

hello.py

1 conflict | Prev | Next | Mark as resolved

```
print("Hi")
print("Hello world")
print("Yes")
print("KK")
```

add some code to hello.py #2

Resolving conflicts between `feature/index` and `main` and committing changes → `feature/index`

This merge commit will be associated with bhattarabhorn.wa@kmitl.ac.th.

Commit merge

1 conflicting file

hello.py
hello.py

hello.py ✓ Resolved

```
print("Hi")
print("Hello world")
print("Yes")
print("KK")
```

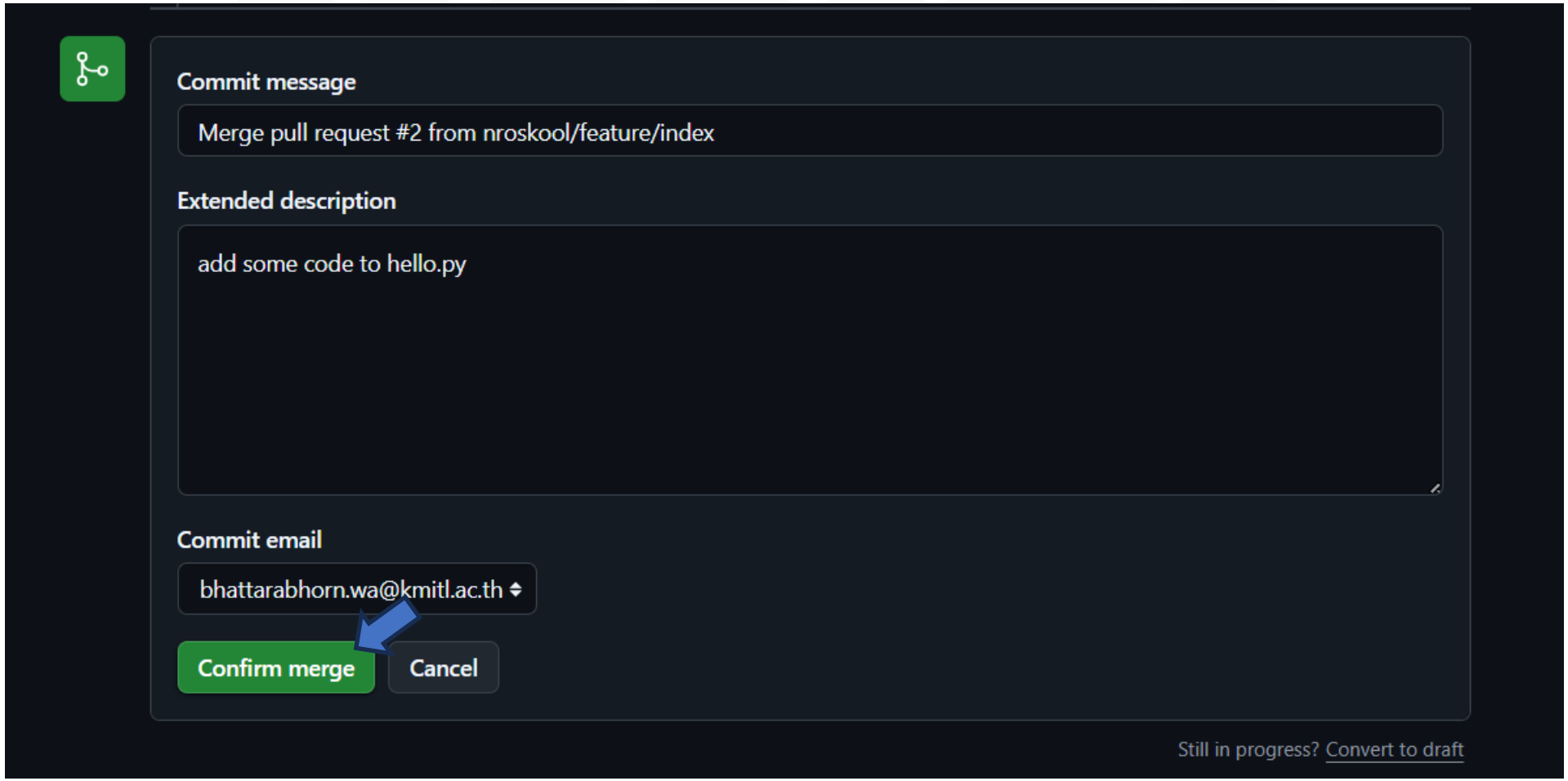
No conflicts with base branch
Merging can be performed automatically.

Merge pull request

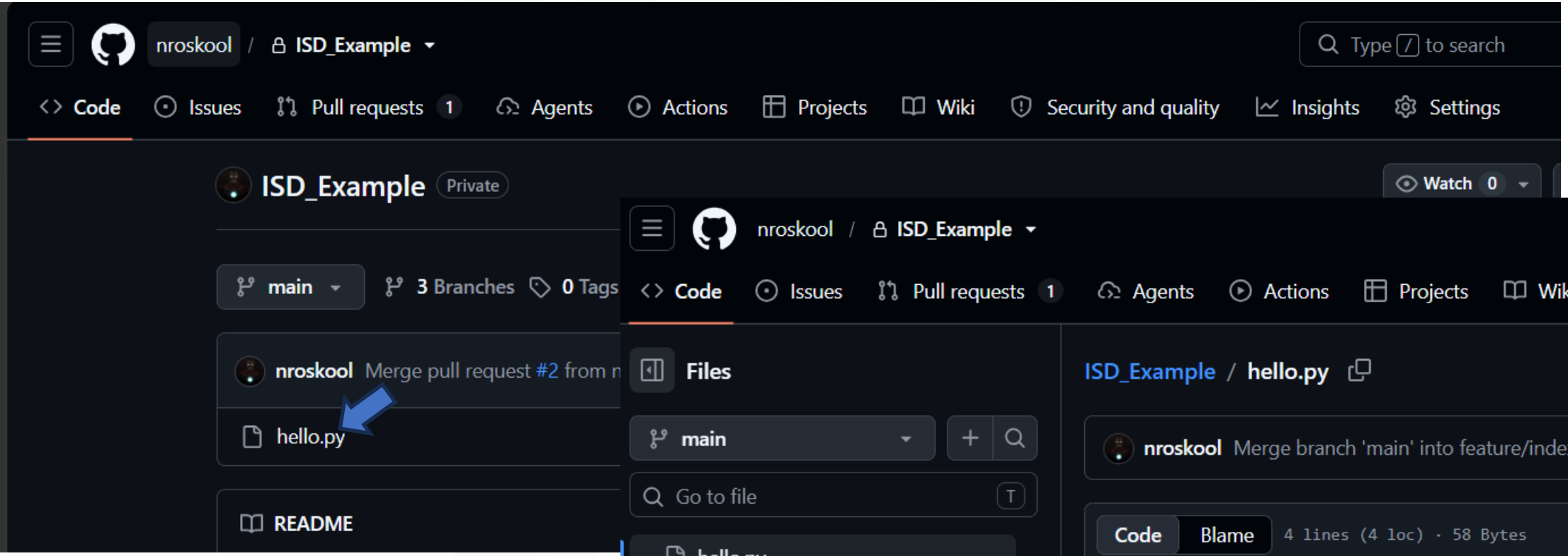
You can also merge this with the command line. View command line instructions.

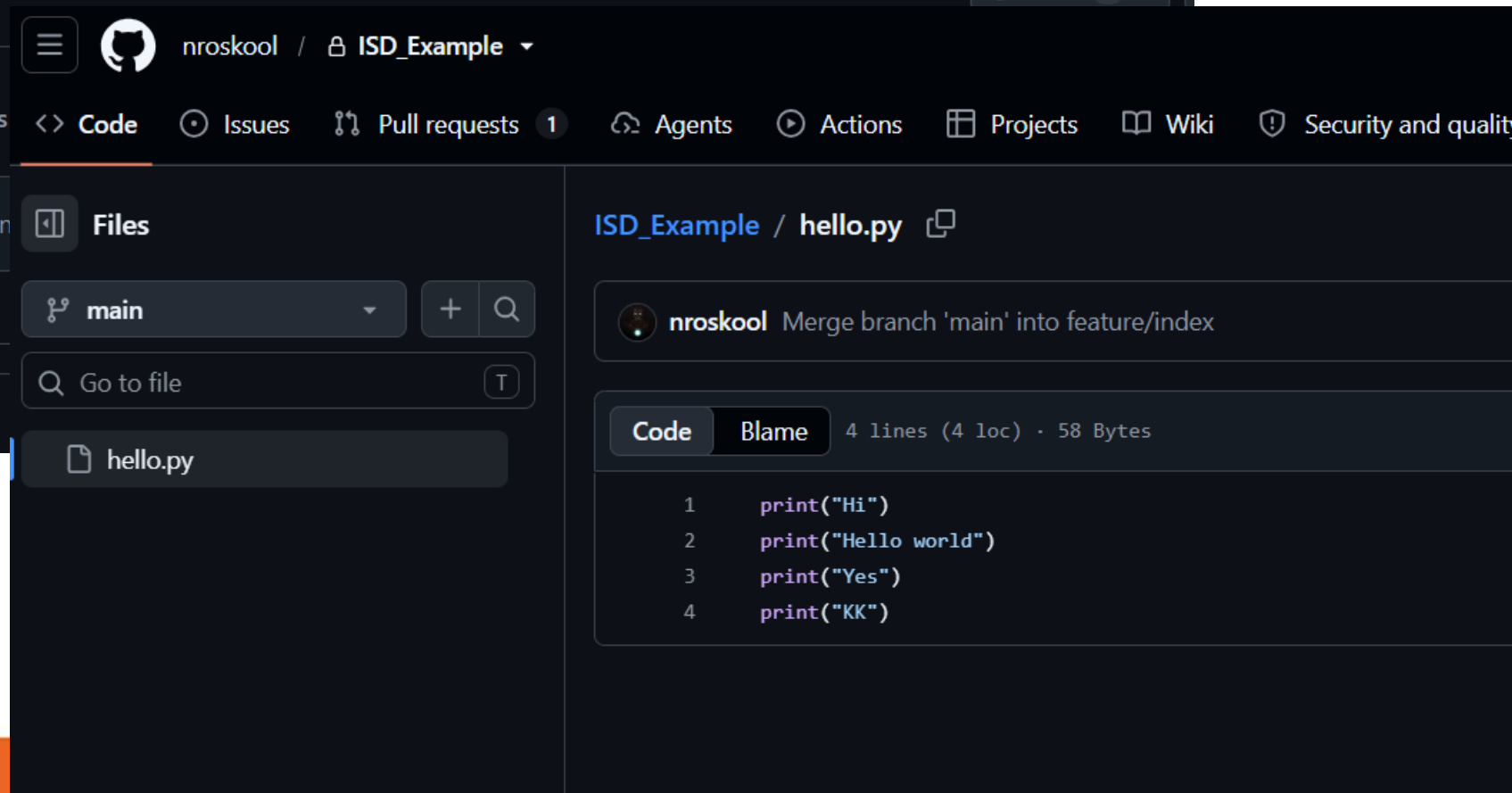
Still in progress? Convert to draft

Feature branch



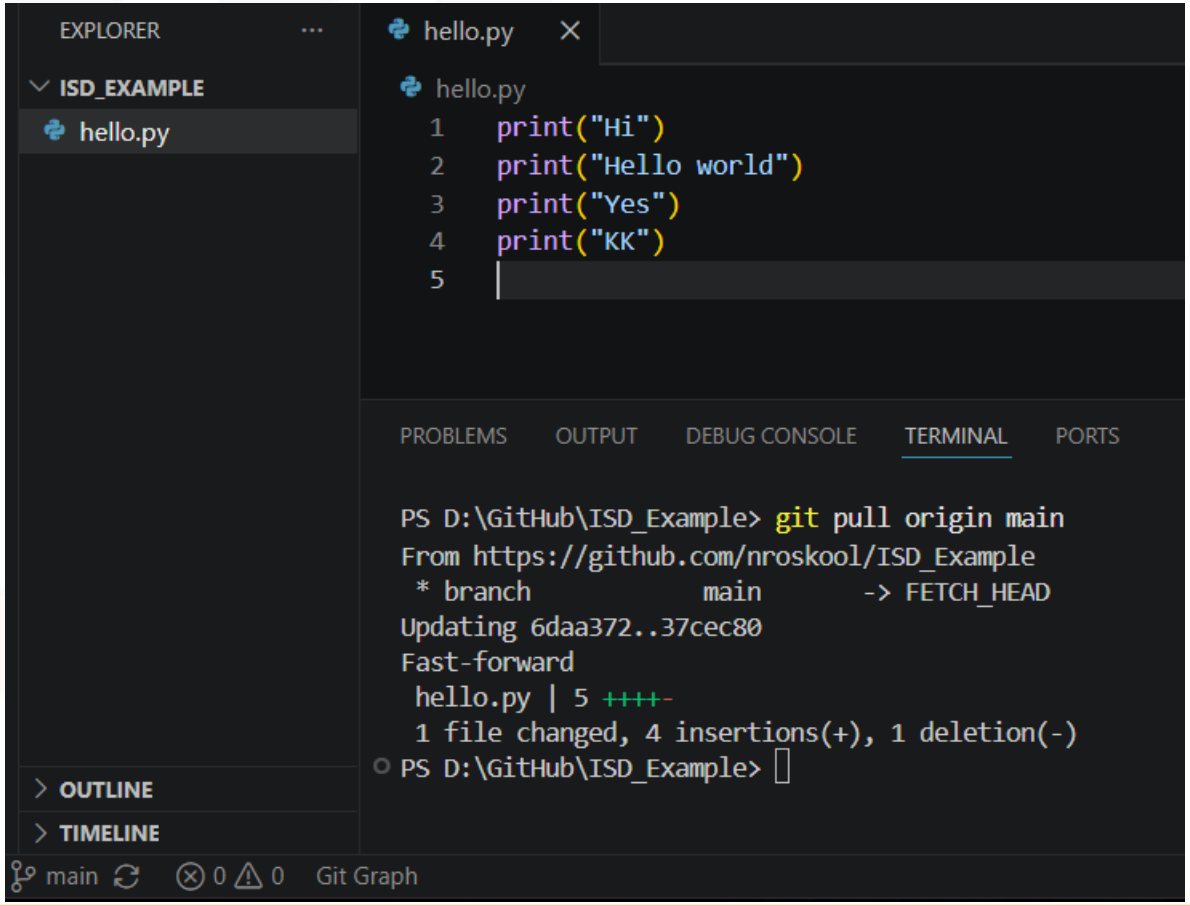
Feature branch





Feature branch

กลับมาที่ vscode ตรง terminal การสลับกลับมาที่ main branch >> **git checkout main**

git pull origin main เพื่อให้ main ใน local repo มีโค้ดและไฟล์เท่ากับ main ใน rermote repo

ถ้าจะทำอะไรต่อ หรือแก้งานใหม่เราก็เริ่ม Slide 28 สร้าง branch ใหม่ สำหรับจัดการต่างๆ.....