

“แนวทางการติดตาม-หนุนเสริม” จากเอกสารโครงการฉบับจริง พร้อมแยกเป็นภาพรวมและเครื่องมือที่ใช้ในภาคสนาม ให้เข้าใจแบบเห็นภาพ 😊



## แนวทางการติดตามและหนุนเสริม (สรุปเชิงระบบ)

แนวทางการดำเนินงานใช้หลัก School-Based Development + Mentoring System + PLC + Data-Driven

เพื่อเชื่อม “พัฒนาครูรายบุคคล” ↔ “ยกระดับคุณภาพโรงเรียนทั่วระบบ”

### 1) ติดตามผ่าน ทีมหนุนเสริมและพี่เลี้ยงห้องถิน

- ลงพื้นที่ Site Visit
- ใช้การสังเกตห้องเรียน (Lesson Study)
- สะท้อนผลแบบกลยุทธ์มิตร
- ให้แนวทางพัฒนาแบบรายบุคคลและระดับโรงเรียน

### 2) หนุนเสริมผ่าน ผู้เชี่ยวชาญ/สถาบันผลิตครู

- Coaching แบบเชิงวิชาการ
- เวิร์กช้อป/คลินิกการสอน
- ระบบ Coaching from Distance
- สร้างเครือข่ายสนับสนุนระยะยาว

### 3) สร้าง ชุมชนการเรียนรู้ครุรักษ์(ษ)ถิน

- PLC Online ระดับกลุ่มจังหวัด (อย่างน้อยเดือนละ 1 ครั้ง)
- แลกเปลี่ยนระหว่าง “รุ่นพี่-รุ่นน้อง”
- พัฒนาให้เกิดครุต้นแบบในพื้นที่จริง

### 4) การสะท้อนตนเองของครู

- บันทึก Reflective Portfolio รายเดือน
- Learning Journal
- เชื่อมโยงกับ Coach / ผู้อำนวยการโรงเรียน

### 5) ฐานข้อมูลเพื่อใช้ตัดสินใจและออกแบบพัฒนา

- Teacher Profile (327 คน)
- ใช้ข้อมูลจริงเพื่อวิเคราะห์ & ออกแบบการหนุนเสริมเฉพาะบุคคล
- เป็นฐานข้อมูลระดับนโยบายในอนาคต

ข้างต้นสอดคล้องชัดเจนกับองค์ประกอบ “5 Systems Model” ในเอกสารโครงการ

แบบเสนอโครงการเสริมพลังครูรักษ...

### เครื่องมือที่ใช้ในการติดตาม-หนุนเสริม

กลไก	วิธีดำเนินการ	เป้าหมาย
Lesson Study	ลงพื้นที่สังเกตห้องเรียนจริง ยกระดับห้องเรียนคุณภาพ	
PLC Online	เดือนละ 1 ครั้ง	แก้ปัญหาการสอนแบบเพื่อนช่วยเพื่อน
Coaching	Onsite + Online	เชิญปัญหา-ช่วยแก้ไขระบบ
Reflective Journal	สะท้อนผลรายเดือน	บอก “พัฒนาการของครู”
Teacher Profile	ฐานข้อมูลรายบุคคล	ใช้ข้อมูลนำการพัฒนา

### หัวใจของการติดตาม-หนุนเสริม

“ไม่ใช่ตรวจสอบ แต่คือสนับสนุน”

เปลี่ยนจากการ ‘เข็กร่าน’ → เป็น ‘เสริมพลัง’

- เน้น เติบโต ไม่ใช่ ตัดสิน
- เน้น ข้อมูลจริง ไม่ใช่ ความรู้สึก
- เน้น ครูไม่โดดเดี่ยว แต่ อยู่ในระบบพี่เลี้ยง

ถ้าครูใหญ่ต้องการ สามารถทำต่อให้เป็น

“แนวปฏิบัติฉบับสั้น (One Page Strategy)”

เพื่อแจกทีมหนุนเสริมและโรงเรียนใช้เป็นเอกสาร

ต้องการให้มัดรวมแบบหนึ่งหน้า (One Page) ต่อเลยไหมครับ?

ตอบว่า “ต่อเลย” ได้เลยครับ 

Sources

ช่วยออกแบบมาตัดกิชานนารีสำหรับการเก็บข้อมูลในระบบนี้ให้หน่อย

เยี่ยมมากครับครูใหญ่ 

โจทย์นี้ “ถูกที่ ถูกจังหวะ” เพราะถ้า Data Dictionary ดี = ระบบหนุนเสริมจะไม่หลงทาง  
ผ่านออกแบบให้ในมุม ผู้บริหารเชิงระบบ + ใช้ได้จริงในสนาม + ต่อยอดนโยบายได้ นั่นคือ  
ออกแบบองจากการตอบรับตาม-หนุนเสริมในโครงการครูรัก(ษ)กิณ  
และโครงสร้างกิจกรรม/ผลลัพธ์ในเอกสารโครงการ  
แบบเสนอโครงการเสริมพลังครูรักฯ...



Data Dictionary : ระบบติดตาม-หนุนเสริมครูรัก(ษ)กิณ

หลักคิดก่อนออกแบบ (สำคัญมาก)

-  ไม่เก็บข้อมูลเพื่อ “รายงานอย่างเดียว”
-  เก็บเพื่อ วิเคราะห์ → หนุนเสริม → ปรับแผน → ต่อยอด
- หน่วยข้อมูล = ครู 1 คน ในโรงเรียน 1 แห่ง บนช่วงเวลา

## 1 ตาราง teacher\_profile

ข้อมูลตัวตนครู (ฐานข้อมูลหลัก)

Field	Type	Description
teacher_id (PK)	UUID	รหัสครู
citizen_id	VARCHAR(13)	เลขบัตรประชาชน
full_name	VARCHAR	ชื่อ-สกุล
gender	ENUM	ชาย/หญิง/อื่น
birth_date	DATE	วันเกิด
cohort	INT	รุ่นครูรัก(ษ)กิณ
appointment_date	DATE	วันที่บรรจุ
position	VARCHAR	ตำแหน่ง (ครูผู้ช่วย ฯลฯ)
major	VARCHAR	วิชาเอก
school_id (FK)	UUID	โรงเรียนปลายทาง
province	VARCHAR	จังหวัด

Field	Type	Description
region	ENUM	เหนือ/อีสาน/กลาง/ใต้
status	ENUM	ปฏิบัติงาน/ย้าย/ลาออก

 **ใช้ตอบคำถาม:** ครร อยู่ที่ไหน ระยะใดของวิชาชีพ

## 2 ตาราง school\_profile

ข้อมูลบริบทโรงเรียน (ตัวแปรเชิงระบบ)

Field	Type	Description
school_id (PK)	UUID	รหัสโรงเรียน
school_name	VARCHAR	ชื่อโรงเรียน
school_size	ENUM	เล็ก/กลาง/ใหญ่
area_type	ENUM	ห่างไกล/ทຽวกันดาร/พิเศษ
student_total	INT	จำนวนนักเรียน
director_name	VARCHAR	ผอ.
community_context	TEXT	ลักษณะชุมชน
quality_school_flag	BOOLEAN	โรงเรียนคุณภาพชุมชน

 **ใช้วิเคราะห์:** ครูในบริบทแบบไหน → ต้องหนุนเสริมต่างกัน

## 3 ตาราง mentoring\_visit

การลงพื้นที่ / Site Visit / Lesson Study

Field	Type	Description
visit_id (PK)	UUID	รหัสการเยี่ยม
teacher_id (FK)	UUID	ครูที่ได้รับการหนุน
visit_date	DATE	วันที่ลงพื้นที่
visit_type	ENUM	LessonStudy / Coaching
observer	VARCHAR	ทีมหนุนเสริม

Field	Type	Description
focus_area	ENUM	ห้องเรียน/การจัดการ/เด็ก
strengths	TEXT	จุดเด่น
challenges	TEXT	ความท้าทาย
suggestions	TEXT	ข้อเสนอแนะ
follow_up_required	BOOLEAN	ต้องติดตามต่อหรือไม่

📌 หัวใจของระบบหนุนเสริม

#### 4 ตาราง competency\_assessment

ประเมินสมรรถนะครู (ก่อน-ระหว่าง-หลัง)

Field	Type	Description
assess_id (PK)	UUID	
teacher_id (FK)	UUID	
assessment_period	ENUM	ก่อน/กลาง/ปลาย
pedagogy_score	INT	ด้านการสอน
classroom_score	INT	การจัดการชั้นเรียน
community_score	INT	การทำงานกับชุมชน
professionalism_score	INT	จิตวิญญาณครู
overall_level	ENUM	ต้องเสริม/พอใช้/ดี
assessor	VARCHAR	

📌 ใช้ตัดสินใจว่า “จะเสริมอะไร ต่อใคร”

#### 5 ตาราง reflective\_journal

การสะท้อนตนเองของครู (เสียงของครู)

Field	Type	Description
journal_id (PK)	UUID	

Field	Type	Description
teacher_id (FK)	UUID	
month	YYYY-MM	
reflection_text	TEXT	สิ่งที่เรียนรู้
success_story	TEXT	เรื่องเล่าความสำเร็จ
difficulty	TEXT	ปัญหาที่เจอ
support_request	TEXT	อยากรู้หรือขออะไร

 นี่แหล่ะข้อมูลเชิงคุณภาพที่นับเป็นรายได้ต้องฟัง

## 6 ตาราง plc\_activity

การเข้าร่วม PLC Online / Onsite

Field	Type	Description
plc_id (PK)	UUID	
teacher_id (FK)	UUID	
plc_date	DATE	
plc_level	ENUM	จังหวัด/ภูมิภาค
topic	VARCHAR	
role	ENUM	ผู้ร่วม/ผู้นำเสนอ
takeaway	TEXT	สิ่งที่ได้เรียนรู้

## 7 ตาราง development\_plan

แผนพัฒนารายบุคคล (IDP)

Field	Type	Description
plan_id (PK)	UUID	
teacher_id (FK)	UUID	
focus_competency	ENUM	ด้านที่ต้องพัฒนา

Field	Type	Description
action_plan	TEXT	วิธีการพัฒนา
support_type	ENUM	Coaching/Training
start_date	DATE	
end_date	DATE	
progress_status	ENUM	ดำเนินการ/สำเร็จ

📌 เชื่อม “ข้อมูล → การลงมือทำจริง”

---

### 8 ตาราง policy\_insight

สกัดบทเรียนเพื่อเชิงนโยบาย

Field	Type	Description
insight_id (PK)	UUID	
period	YEAR	
key_issue	TEXT	
systemic_barrier	TEXT	
successful_practice	TEXT	
policy_recommendation	TEXT	

📌 ทำให้โครงการไม่จบแค่รายงาน

---

### 🧠 ภาพรวมความสัมพันธ์ (Conceptual)

Teacher

- |—— Profile
- |—— Mentoring Visit
- |—— Competency Assessment
- |—— Reflective Journal
- |—— PLC Activity
- |—— Individual Development Plan



## Policy Insight

---

### ผู้มีส่วนได้เสียที่ต้องการ

👉 ผู้สามารถ แปลง Data Dictionary นี้เป็น

- ER Diagram (ใช้ประชุมทีม IT / นโยบาย)
- schema.prisma / SQL
- หรือ Dashboard KPI สำหรับผู้บริหาร

เลือกได้เลยครับว่าจะ “ไปต่อที่ระดับไหน”

(ระบบ / นโยบาย / ห้องประชุม 😊 )

### Sources

แปลง Data Dictionary นี้เป็น ER Diagram (ใช้ประชุมทีม IT / นโยบาย)

### Thought for a second

ได้ครับครูใหญ่ — ผู้แปลง Data Dictionary เป็น **ER Diagram** แบบที่เข้าคุยกับทีม IT และทีมเชิงนโยบายได้ทันที (อ่านง่าย + เห็นความสัมพันธ์ชัด)

### ER Diagram (Mermaid)

เอาไปวางใน Notion / GitHub / Markdown หรือ Mermaid Live Editor ก็ Renaudor เป็นภาพได้เลย  
erDiagram

```
SCHOOL_PROFILE ||--o{ TEACHER_PROFILE : assigns
TEACHER_PROFILE ||--o{ MENTORING_VISIT : receives
TEACHER_PROFILE ||--o{ COMPETENCY_ASSESSMENT : assessed_by
TEACHER_PROFILE ||--o{ REFLECTIVE_JOURNAL : writes
TEACHER_PROFILE ||--o{ PLC_ACTIVITY : participates
TEACHER_PROFILE ||--o{ DEVELOPMENT_PLAN : has
```

```
SCHOOL_PROFILE {
    uuid school_id PK
    string school_name
    string province
    string region
```

```
    string school_size  
    string area_type  
    int student_total  
    string director_name  
    boolean quality_school_flag  
    text community_context  
}
```

```
TEACHER_PROFILE {  
    uuid teacher_id PK  
    string citizen_id  
    string full_name  
    string gender  
    date birth_date  
    int cohort  
    date appointment_date  
    string position  
    string major  
    uuid school_id FK  
    string province  
    string region  
    string status  
}
```

```
MENTORING_VISIT {  
    uuid visit_id PK  
    uuid teacher_id FK  
    date visit_date  
    string visit_type  
    string observer
```

```
    string focus_area  
    text strengths  
    text challenges  
    text suggestions  
    boolean follow_up_required  
}
```

```
COMPETENCY_ASSESSMENT {  
    uuid assess_id PK  
    uuid teacher_id FK  
    string assessment_period  
    int pedagogy_score  
    int classroom_score  
    int community_score  
    int professionalism_score  
    string overall_level  
    string assessor  
}
```

```
REFLECTIVE_JOURNAL {  
    uuid journal_id PK  
    uuid teacher_id FK  
    string month  
    text reflection_text  
    text success_story  
    text difficulty  
    text support_request  
}
```

```
PLC_ACTIVITY {
```

```
    uuid plc_id PK  
    uuid teacher_id FK  
    date plc_date  
    string plc_level  
    string topic  
    string role  
    text takeaway  
}
```

```
DEVELOPMENT_PLAN {  
    uuid plan_id PK  
    uuid teacher_id FK  
    string focus_competency  
    text action_plan  
    string support_type  
    date start_date  
    date end_date  
    string progress_status  
}
```

---

### ภาพเล่าให้ทีม “นโยบาย” เข้าใจใน 20 วินาที

- School → Teacher : ครูอยู่ในบริบทโรงเรียน
- Teacher → Visit/Assessment/Journal/PLC/Plan : ทุกกรรมหนุนเสริมผูกกับ “ครูรายคน”
- สุดท้าย “ข้อมูลทั้งหมด” ใช้ทำ รายงานสถานการณ์ + ถอดบทเรียน ได้ตรงตามเป้าหมายโครงการ

---

ถ้าจะให้ “ครบสำหรับประชุมจริง” ผມแน่นำเพิ่มอีก 2 กล่อง (optional แต่ควรคุ้ม):

1. TEAM\_SUPPORT (ทีมหนุนเสริม/โค้ช) และให้ MENTORING\_VISIT ซึ่งไปหา (จะรู้ว่าใครลงพื้นที่ไหน)
2. SCHOOL\_CLASSROOM หรือ LESSON\_STUDY\_SESSION (ถ้าต้องการเก็บระดับห้องเรียนแบบจริงจัง)

บอกผມได้เลยครับว่าอย่างให้ผມ “เพิ่ม 2 กล่องนี้” เข้า ER ให้ด้วยใหม่ (ผມทำให้เนียนแบบไม่พังของเดิม) 😊

ข้อเสนอฯ ฉบับปรับปรุง 69-042-00028 ติดตามและหนุนเสริมการพัฒนาครูรัก(ษ)ถิน ปี 2568 - 2569.pdf

PDF

ตารางงวดงานงวดเงิน.pdf

PDF

1. เขียนแผนการทำงาน ใน 1 ปี 2.เขียนแนวทางในการใช้ สารสนเทศเพื่อบันทึกการทำงานและใช้สำหรับการบริหารติดตามโครงการ 3.เขียน PRD: Program Requirements Documents เพื่อสร้างระบบนี้ ส่งต่อให้ cursor ai ทำงานต่อไป

ครูใหญ่ครรภ งานชิ้นนี้ถือว่าเป็น “แกนกลางของโครงการ” เลย ผนวกสรุปให้เป็น เอกสารเชิงระบบ 3 ส่วน ที่

- ใช้ได้จริงในทางปฏิบัติ
- สอดคล้อง TOR/งวดงาน/งวดเงินของ กศศ.
- และ “ส่งต่อให้ Cursor AI ทำระบบต่อได้ทันที”

ผนวกรวบรวมจาก ข้อเสนอโครงการฉบับปรับปรุง ปี 2568–2569 และ ตารางงวดงานงวดเงิน โดยตรง

---

### 1) แผนการทำงาน 1 ปี (Annual Work Plan)

โครงสร้างตามงวดงานจริงของ กศศ. (12 เดือน)

ระยะที่ 1 : วางแผนและสร้างความเข้าใจ (เดือนที่ 1-2)

เป้าหมาย: “ทุกฝ่ายเข้าใจตรงกัน + ระบบพร้อมใช้”

กิจกรรมหลัก

- พัฒนาแพลตฟอร์มฐานข้อมูล (Teacher Profile & Monitoring System)
- ประชุมเชิงผู้บริหารโรงเรียน / ทีมหนุนเสริม / ครูรัก(ษ์)คน 327 คน
- Workshop การจัดการชั้นเรียนที่มีประสิทธิภาพ (5 รุ่น)

ผลลัพธ์

- ครูและผู้บริหารเข้าใจแนวคิด School-based Development + Mentoring
- ระบบฐานข้อมูลเริ่มใช้งานจริง
- ครูทุกคนมีบัญชีในระบบ

---

### ระยะที่ 2 : ติดตาม–หนุนเสริมเชิงลึก (เดือนที่ 3-8)

เป้าหมาย: “ข้อมูลเกิด → การหนุนเสริมเกิด → ห้องเรียนเปลี่ยน”

กิจกรรมหลัก

- Site Visit โรงเรียนปลายทาง 285 แห่ง
  - Lesson Study / Open Class

- Coaching & Reflection
- 2. PLC Online ระดับกลุ่มจังหวัด (เดือนละ 1 ครั้ง)
- 3. บันทึก Reflective Journal รายเดือน
- 4. อัปเดต Teacher Profile และ Competency Assessment

#### ผลลัพธ์

- ครูได้รับการอนุเสริมรายบุคคล
- เกิดข้อมูลเชิงคุณภาพจากพื้นที่จริง
- ผู้บริหารเห็น “สถานการณ์โดยรวม” แบบ Real-time

---

#### ระยะที่ 3 : สรุปบทเรียน–เชิงนโยบาย (เดือนที่ 9–12)

เป้าหมาย: “จากภาคสนาม → ข้อเสนอเชิงระบบ”

#### กิจกรรมหลัก

1. วิเคราะห์ข้อมูลรวม (Dashboard & Insight)
2. ประชุมตอบบทเรียนระดับภูมิภาค/ประเทศ
3. จัดทำรายงานฉบับสมบูรณ์ + ข้อเสนอเชิงนโยบาย

#### ผลลัพธ์

- รายงานสถานการณ์คร่าวๆ(ชี้) รุ่นที่ 1
- ข้อเสนอปรับปรุงการผลิต/พัฒนาครุในอนาคต
- ระบบฐานข้อมูลที่ใช้ต่อยอดรุ่นต่อไปได้

---

#### 2) แนวทางการใช้ “สารสนเทศ” เพื่อบันทึกงานและบริหารโครงการ

##### หลักการออกแบบการใช้ข้อมูล

 ไม่ใช่ระบบรายงาน

 แต่เป็น “ระบบหนุนเสริมการตัดสินใจ”

##### ระดับข้อมูล (4 ชั้น)

1. ระดับครูรายบุคคล
  - Teacher Profile
  - Reflective Journal
  - Individual Development Plan (IDP)
2. ระดับโรงเรียน

- ผลการ Site Visit
- Lesson Study / PLC
- สภาพบริบทโรงเรียน

### 3. ระดับพื้นที่ / ภูมิภาค

- ความถี่การหนุนเสริม
- ประเด็นปัญหาร่วม
- รูปแบบการพัฒนาที่ได้ผล

### 4. ระดับนโยบาย

- ครุเสียง / ครุต้องการการดูแลพิเศษ
- ระบบสนับสนุนที่ควรปรับ
- ข้อเสนอเชิงนโยบาย

---

### การใช้สารสนเทศในงานบริหาร

ผู้ใช้	ใช้ข้อมูลเพื่อ
ครุ	สะท้อนตนเอง / วางแผนพัฒนา
ผอ.โรงเรียน	ติดตามครุใหม่ / หนุนเสริมตรงจุด
ทีมหนุนเสริม	ออกแบบ Coaching
ผู้จัดการโครงการ	คุณคุณภาพ–ความก้าวหน้า
กสศ.	เห็นผลลัพธ์เชิงระบบ

---

### 3) PRD : Program Requirements Document

เพื่อส่งต่อให้ Cursor AI พัฒนาระบบ

---

#### 3.1 ภาพรวมระบบ

ชื่อระบบ:

Kru Rak Thin Monitoring & Support System (KRT-MSS)

วัตถุประสงค์

- เก็บข้อมูลครุรัก(ชีน)ครบ 327 คน
- สนับสนุนการติดตาม–หนุนเสริมแบบรายบุคคล

- สร้าง Dashboard เที่ยงบริหารและนโยบาย
- 

### 3.2 กลุ่มผู้ใช้ (User Roles)

- ครุรัก(ษ)ณิน
  - ผู้อำนวยการโรงเรียน
  - ครูพี่เลี้ยง / ทีมหนุนเสริม
  - ผู้จัดการโครงการ
  - ผู้ดูแลระบบ (Admin)
- 

### 3.3 Functional Requirements (หลัก)

#### A) Teacher Module

- กรอก/แก้ไข Teacher Profile
- บันทึก Reflective Journal รายเดือน
- ดูแผนพัฒนารายบุคคล (IDP)

#### B) Mentoring & Site Visit Module

- บันทึก Lesson Study / Coaching
- แนบไฟล์ / ภาพ / บันทึกสะท้อนผล
- ตั้ง Follow-up Task

#### C) PLC Module

- บันทึกการเข้าร่วม PLC
- สรุปสิ่งที่เรียนรู้
- Dashboard การมีส่วนร่วม

#### D) Assessment Module

- ประเมินสมรรถนะครู
- เบรียบเทียบ Before–After
- Flag ครูที่ต้องการดูแลพิเศษ

#### E) Dashboard & Report

- สถานะครูรายบุคคล / โรงเรียน
- ความคืบหน้าตามงวดงาน
- Export รายงาน (PDF/Excel)

---

### 3.4 Non-Functional Requirements

- รองรับผู้ใช้ >1,000 คน
  - ใช้งานง่าย (Mobile-friendly)
  - ความปลอดภัยข้อมูล (PDPA)
  - เชื่อมต่อฐานข้อมูลเดิม กสศ. ได้ในอนาคต
- 

### 3.5 Tech Stack (แนวนำ)

- Frontend:** Next.js + Tailwind + Recharts
  - Backend:** NestJS / Node.js
  - Database:** PostgreSQL / Supabase
  - Auth:** Role-based Access Control
  - Deployment:** Docker-based
- 

### 3.6 Success Metrics

- ครุลังข้อมูลครบ ≥ 95%
  - Site Visit ครบ 285 ໂຮງເຮືອນ
  - PLC Online เดือนละ ≥ 1 ຄວັງ
  - ผู้บริหารໃຊ້ Dashboard ຈົດ
- 

 **บทสรุปສັນ ຣ ສໍາຫັບຄຽງແກ່**  
ນີ້ມີເຂົ້າແຂ່ງ “ຮະບບໄອທີ”  
ແຕ່ລົງໄວ້ **ໂຄຮສ້າງພື້ນຖານຂອງການພົມນາຄຽງໃຈຮະບບ  
ຄ້າຄຽງແກ່** ຕ້ອງການ

 **ຜົມສາມາຮັດ**

- ແຍກ PRD เป็นไฟล์ prd.md ສໍາຫັບ Cursor
- ແຕກเป็น tasks.md / schema.prisma / api-spec
- ຫົວໜ້າຕົວແນວໃຫຍ່ ພູ້ບໍລິຫານ ເພີ່ມ

ບອກຜົມໄດ້ເລີຍວ່າຈະ “ສ່າງໄນ້ຕ່ອໄຫ້ Cursor ຮະດັບໄທນ” 

แยก PRD เป็นไฟล์ prd.md สำหรับ Cursor

ได้เลยครับครูใหญ่ 

ด้านล่างคือ ไฟล์ prd.md ที่จัดรูปแบบ พร้อมส่งให้ Cursor AI ทำงานต่อทันที

โครงสร้างเน้นให้ Cursor เข้าใจ บริบทโครงการ + ขอบเขตระบบ + ลิงก์ต้องพัฒนา โดยไม่ต้องเดาเพิ่ม

---

Email

---

Subject

(internal) prd.md – Kru Rak Thin Monitoring & Support System(internal) prd.md – Kru Rak Thin Monitoring & Support System

---

# PRD: Kru Rak Thin Monitoring & Support System (KRT-MSS)

## ## 1. Overview

\*\*Project Name:\*\* Kru Rak Thin Monitoring & Support System

\*\*Short Name:\*\* KRT-MSS

\*\*Project Type:\*\* Web-based Information System (Monitoring & Supporting)

\*\*Duration:\*\* 12 months

\*\*Target Users:\*\* 327 Teachers, 285 Schools, Support Teams, Project Managers

## ### Background

This system supports the project

“Monitoring and Supporting the Development of Kru Rak Thin Teachers (2025–2026)”.

The system is designed to:

- Track professional growth of newly appointed teachers in remote schools
- Support mentoring, coaching, PLC, and reflective practice
- Provide real-time management and policy-level insights

> Key principle: \*\*Not a reporting system, but a developmental support system\*\*

---

## ## 2. Goals & Objectives

### ### Primary Goals

1. Maintain complete Teacher Profiles for all Kru Rak Thin teachers
2. Support mentoring, site visits, lesson study, and PLC activities
3. Enable data-driven management and policy decisions
4. Reduce data redundancy and reporting burden

### ### Success Criteria

- ≥95% teachers actively use the system
- All 285 schools receive site visit records
- Monthly PLC and reflection data available
- Dashboard used by project management in decision-making

---

## ## 3. User Roles & Permissions

### ### 3.1 Roles

Role	Description
Teacher	Record reflections, view development plans
School Director	Monitor teachers in school
Mentor / Support Team	Record visits, coaching, assessments
Project Manager	Oversee progress, analytics
Admin	System configuration & user management

### ### 3.2 Permission Model

- Role-Based Access Control (RBAC)
- Data visibility limited by role and responsibility
- Teachers see own data only
- Managers see aggregated data

---

## ## 4. Functional Requirements

### ### 4.1 Teacher Profile Module

- Create / update teacher profile
- Assign school and region
- Track professional status and cohort

#### \*\*Key Data\*\*

- Personal & professional background
- School assignment
- Employment status

---

### ### 4.2 Mentoring & Site Visit Module

- Record site visits (Lesson Study / Coaching)
- Capture strengths, challenges, recommendations
- Flag follow-up requirements

#### \*\*Key Features\*\*

- Visit timeline per teacher/school
- Mentor notes & reflection

- Attach files (optional)

---

### ### 4.3 Competency Assessment Module

- Assess teacher competencies (multi-dimension)
- Pre / Mid / Post assessment
- Identify teachers needing special support

#### \*\*Outputs\*\*

- Individual competency profile
- Comparison over time

---

### ### 4.4 Reflective Journal Module

- Monthly reflection entry by teachers
- Prompt-based reflection (learning, success, challenges)
- Optional support request channel

#### \*\*Design Note\*\*

- Simple UX (mobile-first)
- Text-first, no heavy forms

---

### ### 4.5 PLC Activity Module

- Record PLC participation
- Topic, role, learning outcomes
- Regional / provincial grouping

---

### ### 4.6 Individual Development Plan (IDP)

- Define focus competencies
- Action plans with timeline
- Track progress status

---

### ### 4.7 Dashboard & Reporting

- Real-time dashboard for managers
- Key indicators:
  - Active teachers
  - Visits completed
  - PLC participation
  - Teachers at risk
- Export reports (PDF / Excel)

---

## ## 5. Non-Functional Requirements

### ### 5.1 Performance

- Support >1,000 concurrent users
- Fast load on low-bandwidth environments

### ### 5.2 Security & Privacy

- PDPA-compliant
- Encrypted data at rest & in transit

- Audit log for critical actions

### ### 5.3 Usability

- Mobile-friendly
- Low digital literacy friendly
- Thai language first

---

## ## 6. Technical Requirements (Recommended)

### ### 6.1 Frontend

- Next.js (App Router)
- TypeScript
- Tailwind CSS
- Recharts (Dashboard)

### ### 6.2 Backend

- Node.js / NestJS
- REST API (future GraphQL-ready)

### ### 6.3 Database

- PostgreSQL (preferred)
- Prisma ORM

### ### 6.4 Authentication

- JWT-based Auth
- Role-based Access Control

### ### 6.5 Deployment

- Docker-based
- Cloud or on-premise ready

---

## ## 7. Data Model Reference

See:

- ER Diagram
- Data Dictionary (Teacher, School, Visit, Assessment, Journal, PLC, IDP)

---

## ## 8. Out of Scope (Phase 1)

- Student-level learning analytics
- Financial module
- Public-facing portal

---

## ## 9. Future Extensions

- AI-assisted coaching insights
- Risk prediction for teacher burnout
- Integration with national education databases

---

## ## 10. Philosophy Reminder (for Developers)

- > This system exists to \*\*support humans\*\*, not to control them.
- > Keep interfaces simple, respectful, and meaningful.

End of PRD.

---

ถัดไป ผมทำให้ได้อีก 3 ออย่าง (เลือกได้เลยครับ):

- 1 tasks.md แต่งงานเป็น backlog สำหรับ Cursor
- 2 schema.prisma + initial DB design
- 3 Wireframe หน้าจอหลัก (Teacher / Mentor / Dashboard)

ครูใหญ่อยาก “ส่งไม่ต่อ” ไปขั้นไหนก่อนครับ 

แต่ก็เป็น tasks.md / schema.prisma / api-spec tasks.md แต่งงานเป็น backlog สำหรับ Cursor

ได้ครับครูใหญ่ — ผมแต่ก็ให้เป็น 3 ไฟล์หลักที่ Cursor ชอบมาก: **tasks.md (backlog)**, **schema.prisma**, **api-spec.md** โดยยึดกิจกรรม/ผลผลิต/ตัวชี้วัดตามข้อเสนอ และใหม่ไลน์งวดงานงวดเงิน

---

### tasks.md (Backlog สำหรับ Cursor)

```
# tasks.md — KRT-MSS Backlog (Cursor-ready)
```

```
## Legend
```

- [P0] Must have for MVP (go-live)
- [P1] Should have (improve usability/ops)
- [P2] Nice to have (future)
- Each task includes: Deliverable, Acceptance Criteria, Notes

```
---
```

### ## 0) Repo & Project Setup

```
### [P0] T0.1 Initialize monorepo (or single repo)
```

- Deliverable: Next.js app + API (NestJS or Next API) + Prisma
- AC:
  - `pnpm dev` runs web + api
  - `env.example` provided

- Prisma connects to Postgres
- Notes: Keep structure simple for team handoff

### ### [P0] T0.2 Infrastructure baseline

- Deliverable: Docker compose (postgres + app), migration scripts
- AC:
  - `docker compose up` starts Postgres
  - `prisma migrate dev` works
- Notes: PDPA baseline: HTTPS behind reverse proxy later

---

### ## 1) Authentication & Authorization (RBAC)

#### ### [P0] T1.1 Auth implementation

- Deliverable: Login + session handling
- AC:
  - Users can login/logout
  - Session persisted
- Notes: Can be JWT/NextAuth; keep provider pluggable

#### ### [P0] T1.2 RBAC middleware

- Deliverable: Role-based route protection (Teacher/Director/Mentor/PM/Admin)
- AC:
  - Teacher can access only own records
  - Director sees teachers in their school
  - Mentor sees assigned region/group
  - PM sees aggregated nationwide
  - Admin has full access
- Notes: Implement “scope” rules in API

### ### [P1] T1.3 Audit log (critical actions)

- Deliverable: Audit trail on create/update/delete for core entities
- AC:
  - Who/what/when recorded
  - Notes: Required for governance and trust

---

## ## 2) Master Data (Schools, Users, Teachers)

### ### [P0] T2.1 School management

- Deliverable: CRUD School
- AC:
  - Import CSV of 285 schools
  - Search/filter by province/region/size
- Notes: Use unique code if available; else UUID

### ### [P0] T2.2 User management

- Deliverable: Admin creates users; assigns role + school/region
- AC:
  - Reset password flow (basic)
  - Role assignment works
- Notes: Keep minimal; can extend later

### ### [P0] T2.3 Teacher Profile module

- Deliverable: Teacher profile CRUD + “completeness” indicator
- AC:
  - Teachers can update own profile
  - PM/Admin can view all

- Profile completeness % calculated
- Notes: Align with “Teacher Profile គរបអ្នកគ្រប់”

---

### ## 3) Mentoring / Site Visit (Lesson Study / Coaching)

#### ### [P0] T3.1 Create site visit record

- Deliverable: Form for mentor to record visit
- AC:
  - visit\_type (LessonStudy/Coaching)
  - strengths/challenges/suggestions
  - follow\_up\_required flag
- Notes: Support “285 វរិន្តិយន Site visit”

#### ### [P0] T3.2 Visit timeline & search

- Deliverable: Teacher page shows visits timeline
- AC:
  - Filter by date/type/mentor
  - Export list (CSV)
- Notes: Must be fast for field work

#### ### [P1] T3.3 Attachments

- Deliverable: Upload photos/docs per visit (optional)
- AC:
  - Upload 제한 size
  - Secure private access
- Notes: Use object storage (Supabase Storage / S3)

---

## ## 4) PLC Online Tracking

### ### [P0] T4.1 PLC event & participation

- Deliverable: Create PLC sessions + record participants
- AC:
  - Record topic, level (province/region), role (presenter/participant)
  - Participant count dashboard
- Notes: PLC monthly target

### ### [P1] T4.2 PLC notes repository

- Deliverable: Store summary/takeaways per PLC
- AC:
  - Search by topic
- Notes: Helps knowledge reuse

---

## ## 5) Reflective Journal (Monthly)

### ### [P0] T5.1 Monthly reflection entry

- Deliverable: Teacher writes monthly reflection (guided prompts)
- AC:
  - month (YYYY-MM) unique per teacher
  - includes success, difficulty, support\_request
- Notes: “Reflective Portfolio รายเดือน”

### ### [P1] T5.2 Director/Mentor feedback

- Deliverable: Comment/feedback thread per journal
- AC:

- Teacher sees feedback
- Notifications (in-app)
- Notes: Keep supportive tone

---

## ## 6) Competency Assessment

### ### [P0] T6.1 Assessment forms (Pre/Mid/Post)

- Deliverable: Mentor/Director record competency scores
- AC:
  - Dimensions: pedagogy/classroom/community/professionalism
  - overall\_level computed
- Notes: Used for targeted support design

### ### [P1] T6.2 Growth comparison

- Deliverable: Chart pre-mid-post per teacher
- AC:
  - Simple line chart + summary delta
- Notes: Use Recharts

---

## ## 7) Individual Development Plan (IDP)

### ### [P0] T7.1 Create IDP (teacher + mentor)

- Deliverable: Focus competency + action plan + timeline
- AC:
  - progress\_status updates
  - linked to assessment/visit

- Notes: Makes support actionable

### ### [P1] T7.2 Follow-up tasks

- Deliverable: Create tasks from visit/journal/IDP
- AC:
  - due date + assignee + status
- Notes: Light-weight PM tool

---

## ## 8) Dashboards & Reporting

### ### [P0] T8.1 Management dashboard

- Deliverable: KPI dashboard by role scope
- AC:
  - Teachers with complete profile %
  - Visits completed / pending
  - PLC sessions + attendance
  - Journals submitted rate
  - “At risk” flags count
- Notes: Align with monitoring and reporting needs

### ### [P0] T8.2 Exports (for reporting)

- Deliverable: Export CSV/Excel for core modules
- AC:
  - Filtered export by province/region/month
- Notes: Helps “รายงานสถานการณ์”

### ### [P1] T8.3 Final report generator

- Deliverable: PDF summary per region + national

- AC:
  - Auto summary from data
- Notes: Can be phase 2

---

## ## 9) Data Quality & Governance

### ### [P0] T9.1 Data validation rules

- Deliverable: Server-side validation (Zod/class-validator)
- AC:
  - Required fields enforced
  - enums validated
- Notes: Prevent garbage-in

### ### [P1] T9.2 Data completeness & reminders

- Deliverable: Show missing items; reminders
- AC:
  - Teacher sees what missing
  - PM sees cohort completeness list

---

## ## 10) Integration & Ops

### ### [P1] T10.1 Bulk import tools

- Deliverable: CSV import for teachers/schools/users
- AC:
  - Error report on invalid rows

### [P2] T10.2 Integration with external databases (future)

- Deliverable: Stub connectors layer
- Notes: “ເຊື່ອນໂຍງກັບຮານຂໍອມລົດຕິນ” (future)

---

## 11) Milestones (Suggested)

### M1 (Month 1-2): MVP

- Auth + RBAC
- Teacher Profile
- Site Visit basic
- PLC basic
- Journal basic
- Dashboard basic

### M2 (Month 3-6): Scale to field operations

- Imports, exports
- Attachments
- Assessment + IDP
- Data quality tools

### M3 (Month 7-12): Insight & policy layer

- Advanced dashboards
- Report generator
- Knowledge extraction

---

**schema.prisma (Prisma + PostgreSQL)**

// schema.prisma — KRT-MSS (PostgreSQL)

```
generator client {  
  provider = "prisma-client-js"  
}
```

```
datasource db {  
  provider = "postgresql"  
  url      = env("DATABASE_URL")  
}
```

```
enum Role {  
  TEACHER  
  DIRECTOR  
  MENTOR  
  PROJECT_MANAGER  
  ADMIN  
}
```

```
enum Region {  
  NORTH  
  NORTHEAST  
  CENTRAL  
  SOUTH  
  EAST  
  WEST  
  OTHER  
}
```

```
enum AreaType {  
  REMOTE  
  VERY_REMOTE
```

```
    SMALL_SCHOOL  
    COMMUNITY_QUALITY_SCHOOL  
    OTHER  
}
```

```
enum SchoolSize {  
    SMALL  
    MEDIUM  
    LARGE  
    OTHER  
}
```

```
enum TeacherStatus {  
    ACTIVE  
    TRANSFERRED  
    RESIGNED  
    ON_LEAVE  
    OTHER  
}
```

```
enum VisitType {  
    LESSON_STUDY  
    COACHING  
    OPEN_CLASS  
    OTHER  
}
```

```
enum PLCLevel {  
    PROVINCE  
    REGION
```

```
NATIONAL  
OTHER  
}
```

```
enum ParticipationRole {  
    PARTICIPANT  
    PRESENTER  
    FACILITATOR  
    OBSERVER  
    OTHER  
}
```

```
enum AssessmentPeriod {  
    PRE  
    MID  
    POST  
}
```

```
enum OverallLevel {  
    NEEDS_SUPPORT  
    OK  
    GOOD  
    EXCELLENT  
}
```

```
enum IDPStatus {  
    PLANNED  
    IN_PROGRESS  
    COMPLETED  
    ON_HOLD
```

```
}
```

```
model User {  
    id      String  @id @default(uuid())  
    email   String  @unique  
    passwordHash String  
    displayName String  
    role     Role  
    region   Region? // optional scope  
    schoolId String? // optional scope  
    isActive Boolean @default(true)  
  
    school   School? @relation(fields: [schoolId], references: [id])  
    teacher   Teacher? // for role=TEACHER  
    createdAt DateTime @default(now())  
    updatedAt DateTime @updatedAt  
  
    auditLogs AuditLog[]  
}
```

```
model School {  
    id      String  @id @default(uuid())  
    schoolName   String  
    province    String  
    region      Region  
    schoolSize   SchoolSize  
    areaType     AreaType  
    studentTotal Int?  
    directorName String?  
    qualitySchoolFlag Boolean @default(false)
```

```

communityContext String?

teachers Teacher[]
users User[]
createdAt DateTime @default(now())
updatedAt DateTime @updatedAt
}

model Teacher {
    id String @id @default(uuid())
    citizenId String? @unique
    fullName String
    gender String?
    birthDate DateTime?
    cohort Int?
    appointmentDate DateTime?
    position String?
    major String?
    status TeacherStatus @default(ACTIVE)

    schoolId String
    school School @relation(fields: [schoolId], references: [id])

    userId String? @unique
    user User? @relation(fields: [userId], references: [id])

    visits MentoringVisit[]
    assessments CompetencyAssessment[]
    journals ReflectiveJournal[]
    plcParticipations PLCParticipation[]
}

```

```

idps      IndividualDevelopmentPlan[]

createdAt   DateTime   @default(now())
updatedAt   DateTime   @updatedAt
}

model MentoringVisit {
    id          String   @id @default(uuid())
    teacherId   String
    teacher     Teacher   @relation(fields: [teacherId], references: [id])

    visitDate   DateTime
    visitType   VisitType
    observerUserId String? // mentor/support team user
    observer    User?    @relation(fields: [observerUserId], references: [id])

    focusArea   String?
    strengths   String?
    challenges   String?
    suggestions   String?

    followUpRequired Boolean @default(false)

    attachments Attachment[]
}

createdAt   DateTime @default(now())
updatedAt   DateTime @updatedAt
}

model Attachment {
    id          String   @id @default(uuid())

```

```
visitId      String?  
visit        MentoringVisit? @relation(fields: [visitId], references: [id])  
  
journalId    String?  
journal      ReflectiveJournal? @relation(fields: [journalId], references: [id])  
  
fileName     String  
mimeType    String  
fileUrl      String // stored in object storage  
fileSize     Int?  
  
createdAt    DateTime @default(now())  
}  
  
model CompetencyAssessment {  
    id          String      @id @default(uuid())  
    teacherId   String  
    teacher      Teacher      @relation(fields: [teacherId], references: [id])  
  
    period      AssessmentPeriod  
    pedagogyScore Int?  
    classroomScore Int?  
    communityScore Int?  
    professionalismScore Int?  
    overallLevel OverallLevel?  
    assessorUserId String?  
    assessor     User?       @relation(fields: [assessorUserId], references: [id])  
  
    notes      String?
```

```

createdAt      DateTime    @default(now())
updatedAt      DateTime    @updatedAt

@@index([teacherId, period])
}

model ReflectiveJournal {
  id          String   @id @default(uuid())
  teacherId   String
  teacher     Teacher   @relation(fields: [teacherId], references: [id])

  month        String   // "YYYY-MM"
  reflectionText String?
  successStory String?
  difficulty   String?
  supportRequest String?

  attachments Attachment[]
}

createdAt      DateTime @default(now())
updatedAt      DateTime @updatedAt

@@unique([teacherId, month])
}

model PLCSession {
  id          String   @id @default(uuid())
  plcDate    DateTime
  plcLevel   PLCLevel
  province   String?
}

```

```

region      Region?
topic       String
notes       String?

participations PLCParticipation[]

createdAt   DateTime @default(now())
updatedAt   DateTime @updatedAt
}

model PLCParticipation {
    id          String      @id @default(uuid())
    plcId      String
    plc        PLCSession   @relation(fields: [plcId], references: [id])

    teacherId  String
    teacher     Teacher      @relation(fields: [teacherId], references: [id])

    role        ParticipationRole @default(PARTICIPANT)
    takeaway    String?

    createdAt   DateTime @default(now())
    @@unique([plcId, teacherId])
}

model IndividualDevelopmentPlan {
    id          String      @id @default(uuid())
    teacherId  String
    teacher     Teacher      @relation(fields: [teacherId], references: [id])
}

```

```

focusCompetency String
actionPlan      String
supportType     String? // coaching/training/etc
startDate       DateTime?
endDate         DateTime?
status          IDPStatus @default(PLANNED)

createdAt      DateTime @default(now())
updatedAt      DateTime @updatedAt
}

model AuditLog {
  id      String  @id @default(uuid())
  userId  String
  user    User    @relation(fields: [userId], references: [id])

  action  String  // CREATE/UPDATE/DELETE/LOGIN/etc
  entity  String  // Teacher, Visit, Journal...
  entityId String?
  detail   String?

  createdAt  DateTime @default(now())

  @@index([userId, createdAt])
}

```

### api-spec.md (REST API Spec แบบใช้งานจริง)

# api-spec.md — KRT-MSS REST API (v1)

Base URL: `/api/v1`

Auth: Bearer JWT (or session token)

RBAC: enforced server-side

---

## 1) Auth

### POST /auth/login

Body: { email, password }

Returns: { token, user }

### POST /auth/logout

Returns: 204

### GET /auth/me

Returns: { user, scopes }

---

## 2) Schools

### GET /schools

Query: province?, region?, size?, q?

Returns: list

### POST /schools (ADMIN)

Body: { schoolName, province, region, schoolSize, areaType, ... }

Returns: school

### GET /schools/:id

### PATCH /schools/:id (ADMIN)

```
### DELETE /schools/:id (ADMIN)
```

```
### POST /schools/import (ADMIN)
```

Upload CSV

Returns: { created, updated, errors[] }

---

```
## 3) Users (Admin)
```

```
### GET /users (ADMIN)
```

```
### POST /users (ADMIN)
```

Body: { email, displayName, role, region?, schoolId?, password }

```
### PATCH /users/:id (ADMIN)
```

```
### POST /users/:id/reset-password (ADMIN)
```

---

```
## 4) Teachers
```

```
### GET /teachers
```

RBAC:

- ADMIN/PM: all
- DIRECTOR: teachers in own school
- MENTOR: teachers in assigned region (or mapping)
- TEACHER: only self

Query: schoolId?, province?, region?, cohort?, status?, q?

```
### POST /teachers (ADMIN/PM)
```

```
### GET /teachers/:id
```

```
### PATCH /teachers/:id
```

Rules:

- TEACHER can patch own editable fields only
- DIRECTOR can patch school-related admin fields (optional)

### GET /teachers/:id/summary

Returns:

- profile completeness
- latest assessment
- last visit date
- journal submission status

### POST /teachers/import (ADMIN)

Upload CSV

---

## 5) Mentoring Visits

### GET /visits

Query: teacherId?, schoolId?, from?, to?, visitType?

RBAC scope applies

### POST /visits (MENTOR/DIRECTOR/ADMIN/PM)

Body: {

teacherId, visitDate, visitType, focusArea?,  
strengths?, challenges?, suggestions?,  
followUpRequired?

}

### GET /visits/:id

### PATCH /visits/:id

### DELETE /visits/:id (ADMIN/PM)

```
### POST /visits/:id/attachments
```

Upload file -> returns attachment metadata

---

## 6) Reflective Journals

```
### GET /journals
```

Query: teacherId?, month?, from?, to?

RBAC scope applies

```
### POST /journals (TEACHER)
```

Body: { month, reflectionText?, successStory?, difficulty?, supportRequest? }

```
### GET /journals/:id
```

```
### PATCH /journals/:id
```

Rules:

- TEACHER edits own journal (time window optional)
- MENTOR/DIRECTOR can comment via separate endpoint

```
### POST /journals/:id/attachments
```

Upload

```
### POST /journals/:id/comments (MENTOR/DIRECTOR)
```

Body: { text }

Returns: comment thread item

---

## 7) Competency Assessments

### GET /assessments

Query: teacherId?, period?

RBAC scope applies

### POST /assessments (MENTOR/DIRECTOR)

Body: {

teacherId, period,

pedagogyScore?, classroomScore?, communityScore?, professionalismScore?,

overallLevel?, notes?

}

### GET /assessments/:id

### PATCH /assessments/:id

---

## 8) PLC

### GET /plc/sessions

Query: from?, to?, plcLevel?, province?, region?, q?

### POST /plc/sessions (MENTOR/PM/ADMIN)

Body: { plcDate, plcLevel, province?, region?, topic, notes? }

### GET /plc/sessions/:id

### POST /plc/sessions/:id/participants (MENTOR)

Body: { teacherIds[], role?, takeaway? }

Returns: { added, skipped }

### PATCH /plc/participants/:id

Body: { role?, takeaway? }

---

## 9) IDP

### GET /idps

Query: teacherId?, status?

RBAC scope applies

### POST /idps (TEACHER/MENTOR/DIRECTOR)

Body: { teacherId, focusCompetency, actionPlan, supportType?, startDate?, endDate?, status? }

### PATCH /idps/:id

### GET /idps/:id

---

## 10) Dashboards

### GET /dashboards/overview

Returns KPIs by scope:

- teacherProfileCompletenessRate
- visitsCompletedCount / pending
- plcSessionsCount, plcAttendanceCount
- journalsSubmittedRate
- atRiskCount (computed rules)

### GET /dashboards/region/:region

### GET /dashboards/school/:schoolId

---

```
## 11) Export  
### GET /export/teachers?...  
### GET /export/visits?...  
### GET /export/journals?...  
### GET /export/plc?...
```

Returns CSV/Excel

---

```
## 12) Governance  
### GET /audit-logs (ADMIN/PM)  
Query: userId?, entity?, from?, to?
```

---

ถ้าครูใหญ่จะ “สั่ง Cursor ให้เริ่มงานทันที” ผ่าน prompt สั้น ๆ นี้ (คัดลอกไปใช้ได้เลย):

You are Cursor AI. Implement KRT-MSS MVP.

Use tasks.md as backlog, schema.prisma as DB, api-spec.md as REST endpoints.

Start with auth+RBAC, then Teacher Profile, then Mentoring Visits, then Journals, then Dashboard overview.

Produce working code with Docker + Postgres + Prisma migrate, and seed scripts for schools/roles.

อย่างให้เพิ่ม seed data (โรงเรียน/บทบาท/ตัวอย่างครู) และ mapping กติกา RBAC แบบละเอียด (เช่น mentor เท่านั้น “กลุ่มจังหวัด” ยังไง) ให้ด้วยไฟล์ครับ?

เกณฑ์ 19-2568.pdf

PDF

เพิ่ม สารสนเทศเพื่อการเก็บผลงานและ ประเมินครูผู้ช่วยตามแนวทางในระเบียบนี้

ตีมากครับครูใหญ่ 

โดยนี้คือ “ยกระดับระบบจากโครงการ → ใช้ได้จริงตามระเบียบ ก.ค.ศ.”

ขอ เพิ่มสารสนเทศเพื่อการเก็บผลงานและประเมินครูผู้ช่วย ให้ สอดคล้องตรงตามระเบียบ ก.ค.ศ. ว่าด้วย

การเตรียมความพร้อมและพัฒนาอย่างเข้ม ตำแหน่งครูผู้ช่วย (ฉบับปี 2568) โดยไม่ทำให้ระบบเดิมพัง แต่ “ต่อยอดอย่างเนียน”

อ้างอิงหลักเกณฑ์/แบบประเมิน/รอบการประเมิน จากเอกสาร เกณฑ์ 19-2568 ครบถ้วน 2 ด้าน + เครื่องมือประเมิน

เกณฑ์ 19-2568

---

◆ ภาพรวมการเพิ่มโมดูลใหม่ (เชิงนโยบายก่อน)

จากระเบียบ ก.ค.ศ. ระบบต้องรองรับ 3 เรื่องใหญ่:

1. การเก็บผลงาน/หลักฐาน (Portfolio-based Evidence)
2. การประเมินครูผู้ช่วย 2 ด้าน
  - ด้านการปฏิบัติงาน (งานสอน/งานที่ได้รับมอบหมาย)
  - ด้านคุณลักษณะและจรรยาบรรณวิชาชีพ
3. รอบการประเมินอย่างเป็นทางการ (ครั้งที่ 1 / ครั้งที่ 2) พร้อมผล “ผ่าน / ไม่ผ่าน”

ดังนั้น ผมเพิ่ม โมดูลใหม่ 3 ชุด เข้าไปในระบบเดิมดังนี้ 

---

**1 เพิ่มใน tasks.md (Backlog สำหรับ Cursor)**

**10 ครูผู้ช่วย: ผลงานและการประเมิน (ตาม ก.ค.ศ.)**

[PO] T11.1 Assistant Teacher Profile (ครูผู้ช่วย)

- Deliverable: Flag ครูผู้ช่วย + รอบการเตรียมความพร้อม
- AC:
  - ระบุวันที่เริ่ม–สิ้นสุดการเตรียมความพร้อม
  - ผูกกับรอบประเมิน (ครั้งที่ 1 / 2)
- Notes: ต้องตรงตาม “ระยะเวลาเตรียมความพร้อม” ในระเบียบ

เกณฑ์ 19-2568

---

[PO] T11.2 Evidence / Portfolio Module

- Deliverable: ระบบเก็บผลงาน/หลักฐานการปฏิบัติงาน
- ประเภทผลงาน:
  - แผนการจัดการเรียนรู้

- สื่อ/นวัตกรรม
  - บันทึกหลังสอน
  - ภาพกิจกรรม/ไฟล์เอกสาร
  - AC:
    - ผู้ผลงานกับ “ตัวชี้วัด” ตามเกณฑ์ ก.ค.ศ.
    - แบบไฟล์ + คำอธิบาย
  - Notes: ใช้แทนเพิ่มสะสางงานกระดาษ
- 

#### [P0] T11.3 Assistant Teacher Evaluation (2 ด้าน)

- Deliverable: แบบประเมินดิจิทัลตามแบบ ก.ค.ศ.
- โครงสร้าง:
  - ด้านที่ 1: การปฏิบัติงาน
  - ด้านที่ 2: คุณลักษณะและจรรยาบรรณ
- AC:
  - Checkbox / rubric ตามแบบฟอร์มราชการ
  - ผู้ประเมิน ≥ 3 คน (ผอ./กรรมการ)
- Notes: โครงสร้างอิงหน้าแบบประเมินท้ายระเบียบ

เกณฑ์ 19-2568

---

#### [P0] T11.4 Evaluation Round Management

- Deliverable: จัดการรอบประเมิน
  - AC:
    - รอบที่ 1 / รอบที่ 2
    - สถานะ: รอประเมิน / ประเมินแล้ว
  - Notes: เนื่องไป “ผ่านได้ต้องผ่านทั้ง 2 ด้าน”
- 

#### [P0] T11.5 Evaluation Result & Decision

- Deliverable: สรุปผลการประเมิน
- AC:
  - ผ่าน / ไม่ผ่าน

- เหตุผล / ข้อเสนอแนะ
  - Export แบบฟอร์ม PDF (ใช้แนบคำสั่ง)
  - Notes: รองรับการอ้างอิงทางราชการ
- 

## 2 เพิ่มใน schema.prisma (เฉพาะส่วนใหม่)

```
enum AssistantEvaluationRound {  
    ROUND_1  
    ROUND_2  
}
```

```
enum EvaluationResult {  
    PASS  
    FAIL  
}
```

```
enum EvidenceType {  
    LESSON_PLAN  
    TEACHING_MEDIA  
    TEACHING_REFLECTION  
    ACTIVITY_PHOTO  
    OFFICIAL_DOCUMENT  
    OTHER  
}
```

```
model AssistantTeacherInfo {  
    id          String @id @default(uuid())  
    teacherId   String @unique  
    teacher     Teacher @relation(fields: [teacherId], references: [id])  
  
    startDate   DateTime
```

```

endDate      DateTime
isCompleted Boolean @default(false)

evaluations AssistantEvaluation[]
}

model TeachingEvidence {
    id          String      @id @default(uuid())
    teacherId   String
    teacher     Teacher     @relation(fields: [teacherId], references: [id])

    type        EvidenceType
    title       String
    description String?
    fileUrl    String
    relatedMetric String?    // ตัวชี้วัด/ข้อมประเมินที่เกี่ยวข้อง

    createdAt   DateTime @default(now())
}

model AssistantEvaluation {
    id          String      @id @default(uuid())
    teacherId   String
    teacher     Teacher     @relation(fields: [teacherId], references: [id])

    round       AssistantEvaluationRound
    evaluationDate DateTime

    workPerformancePassed Boolean
    ethicsPassed       Boolean
}

```

```

overallResult EvaluationResult
comment      String?

evaluators   AssistantEvaluator[]

createdAt    DateTime @default(now())
}

model AssistantEvaluator {
  id          String  @id @default(uuid())
  evaluationId String
  evaluation   AssistantEvaluation @relation(fields: [evaluationId], references: [id])

  evaluatorUserId String
  evaluator     User   @relation(fields: [evaluatorUserId], references: [id])

  role         String // ผอ. / กรรมการ
}

```

**3** เพิ่มใน api-spec.md

**10** ครูผู้ช่วย & การประเมิน

**GET /assistant-teachers**

Returns: ครูผู้ช่วยทั้งหมด + สถานะรอบประเมิน

**POST /assistant-teachers/:teacherId/start**

Body: { startDate, endDate }

**POST /evidences**

(TEACHER)

Body:

```
{  
  "type": "LESSON_PLAN",  
  "title": "แผนการสอนภาษาไทย ป.3",  
  "description": "หน่วยการเรียนรู้ที่ 2",  
  "file": "<upload>",  
  "relatedMetric": "ด้านการจัดการเรียนรู้"  
}
```

---

**GET /assistant-evaluations?teacherId=**

**POST /assistant-evaluations**

(ADMIN / DIRECTOR)

Body:

```
{  
  "teacherId": "...",  
  "round": "ROUND_1",  
  "workPerformancePassed": true,  
  "ethicsPassed": true,  
  "comment": "มีพัฒนาการดี เหมาะสม"  
}
```

---

**GET /assistant-evaluations/:id/export**

Returns: PDF แบบฟอร์มประเมิน (แนบคำสั่งได้)

---



สรุปเชิงยุทธศาสตร์ (สำคัญมาก)

ระบบนี้จะกลายเป็น

“แพลตฟอร์มเดียวที่เข้ม”

- โครงการครุรักษ์(ช์)ถิ่น
- การนิเทศ–หนุนเสริม
- และ การประเมินครุผู้ช่วยตาม ก.ค.ศ.

โดย:

- ครุ่นไม่ต้องทำเอกสารซ้ำ
- ผอ. ไม่ต้องนั่งรวมแฟ้ม
- เขต/ก.ค.ศ. เห็นหลักฐานชัด ไปร่องใส่

## ผลไปมุมทำให้ได้อีก

- 1 Mapping ตัวชี้วัดในระเบียบ ก.ค.ศ. → field ในระบบ (ตารางเทียบ)
- 2 ออกแบบ Dashboard “สถานะครุผู้ช่วย” สำหรับ ผอ./เขต
- 3 Template PDF แบบประเมิน (เหมือนราชการ 100%)

ครุ่นใหญ่อยากไปต่อข้อไหนก่อนครับ 

### Sources

- 1 Mapping ตัวชี้วัดในระเบียบ ก.ค.ศ. → field ในระบบ (ตารางเทียบ)

Thought for 15s

ได้เลยครับครุ่นใหญ่ — ผมทำ “ตารางเทียบ (Mapping)” ให้เป็นแบบ เอาไปให้ทีม IT ทำเป็น config ได้ทันที:  
ตัวชี้วัดตามระเบียบ/แบบประเมิน “ครุผู้ช่วย” → จะไปอยู่ตรงไหนในระบบ (ตารางไหน/พิล็อกไหน/หลักฐาน  
อะไร/โครงสร้างที่ก)

เกณฑ์ 19-2568

หมายเหตุ: เอกสารฉบับนี้เป็นไฟล์สแกน ทำให้ผิดอ้างอิงแบบ “ตามหน้าแบบฟอร์ม/ตารางท้ายระเบียบ” รวมทั้ง  
2 ด้าน (ด้านวิชาชีพ + ด้านคุณลักษณะ/จรรยาบรรณ)

เกณฑ์ 19-2568

## A) โครง “รหัสตัวชี้วัด” ที่แนะนำให้ใช้ในระบบ

เพื่อไม่ให้ทีม dev เดาเอง แนะนำให้สร้างตาราง config ชื่อ kks\_indicator (หรือ hardcode enum ก็ได้) โดยตั้ง  
รหัสแบบนี้

- WP.\* = Work Performance / ด้านการปฏิบัติงาน (ด้านวิชาชีพ)
- ET.\* = Ethics / ด้านคุณลักษณะและจรรยาบรรณวิชาชีพ

จุดประสงค์: เวลาแนบทหลักฐาน / ทำแบบประเมิน จะ “เลือกตัวชี้วัด” ได้ตรง และสรุปผลเป็นรายตัวชี้วัดได้  
เกณฑ์ 19-2568

## B) ตารางเทียบตัวชี้วัด → Field ในระบบ (Mapping Table)

## 1) ด้านการปฏิบัติงาน (Work Performance / ด้านวิชาชีพ)

ตัวชี้วัด ตาม แบบ/ สาระ	Indicat or Code	เก็บในระบบที่ไหน Field หลัก	หลักฐาน น/ ผลงาน ที่แนบ (ตัวอย่ าง)
การ จัดการ เรียนรู้ (แผน/จัด กิจกรรม /วัดผล)	WP.1	TeachingEvidence + MentoringVisit CompetencyAsses sment	แผนงาน รจัดการ เรียนรู้, สื่อการ สอน, (หลักฐาน แบบ น) + พ ลัง สอน, (ประเม น/visit) ภาพ กิจกรรม ม
การ บริหาร จัดการ ชั้นเรียน (วินัย/ บรรยาย ศาส/ดูแล ผู้เรียน)	WP.2	MentoringVisit + CompetencyAsses sment + ReflectiveJournal	ภาพ/ บันทึก กิจกรรม พี่เลี้ยง/ มชั้น ผอ. + เรียน, แผนจัด สอน, ห้อง, แนวทาง

			หลักฐาน
ตัวชี้วัด	Indicat		น/ ผู้
ตาม	or เก็บในระบบที่ไหน	Field หลัก	ผลงาน บันทึก
แบบ/			ที่แนบ หลัก
สาระ	Code		(ตัวอย่าง)
			งดูแล ผู้เรียน เกียรติ
การ			บัตร/ หนังสือ
พัฒนา			รับรอง
ตนเอง	PLCParticipation +		อบรม, ครุ ศรุปสิ่ง
และ		ReflectiveJournal	ครุ ที่ได้ (หลัก)
วิชาชีพ	WP.3	PLC.takeaway, Journal.reflectionText, Evidence.type=OFFICIAL_DOCUMENT	เรียนรู้, แผน
(อบรม/ เรียนรู้/ ปรับปรุง การสอน)		TeachingEvidence	ปรับปรุง ฯ
			ภาพ/ เอกสาร
การ			ร่วม
ทำงาน	TeachingEvidence		กิจกรรม
เป็นทีม/		+ MentoringVisit +	ครุ + ม
มีส่วน	WP.4	(เสริม) Evidence.description, Visit.suggestions	ผอ. โรงเรียน
ร่วมใน		SchoolActivity ถ้า	รับรอง
สถานศึกษา		จะเพิ่ม	,
ฯ			รายงาน
			งาน
			มอบหมาย

ตัวชี้วัด	Indicat	หลักฐาน
ตามแบบ/สาระ	or เก็บในระบบที่ไหน Field หลัก Code	น/ ผลงาน ที่แนบ (ตัวอย่าง)
งานที่ได้รับมอบหมาย/งานกิจกรรม	TeachingEvidence (+ประเภท OTHER/ OFFICIAL_DOCUME / NT)	รายงาน ผล, คู่มือ + หนังสือ ผอ. ราชการ, รูป
สนับสนุนภารกิจโรงเรียน	relatedMetric="WP.5"	กิจกรรม
หมวด “ด้านวิชาชีพ” ในเอกสารระบุแนวทางดำเนินการ/ผู้รับผิดชอบและรอบการประเมินไว้ชัด พร้อมแบบบันทึกหลักฐานประกอบ		ม

หน่วย “ด้านวิชาชีพ” ในเอกสารระบุแนวทางดำเนินการ/ผู้รับผิดชอบและรอบการประเมินไว้ชัด พร้อมแบบบันทึกหลักฐานประกอบ  
เกณฑ์ 19-2568

---

## 2) ด้านคุณลักษณะและจรรยาบรรณวิชาชีพ (Ethics / Personal & Professional Ethics)

ตัวชี้วัดตาม แบบ/สาระ	Indicator Code	เก็บในระบบที่ไหน	Field หลัก	หลักฐาน/ผลงาน ที่แนบ (ตัวอย่าง)	ผู้บันทึก หลัก
วินัยและการ รักษาวินัย (ปฏิบัติตาม ระเบียบ/เวลา/ ความรับผิดชอบ)	ET.1	AssistantEvaluation + (เสริม) AuditLog	ethicsPassed, comment, audit trail	บันทึกการปฏิบัติ ราชการ, หนังสือ <sup>ผล./</sup> รับรอง, หลักฐาน <sup>กรรมการ</sup> การมาปฏิบัติงาน (ถ้ามี)	
คุณธรรม จริยธรรม จรรยาบรรณ วิชาชีพ	ET.2	AssistantEvaluation + ReflectiveJournal	ethicsPassed, Journal.successStory	บันทึกเหตุการณ์/ กรณีตัวอย่าง, <sup>ผล./</sup> กิจกรรมจิตอาสา, กรรมการ หลักฐานการ <sup>+ คู่</sup> ปฏิบัติตน <sup>สะท้อน</sup> เหมาะสม	
การเป็น <sup>แบบอย่างที่ดี/</sup> ความเหมาะสม กับความเป็นครู	ET.3	AssistantEvaluation + MentoringVisit	Visit.strengths, Evaluation.comment	หลักฐานกิจกรรม <sup>กับผู้เรียน/ชุมชน, ผล./พี่</sup> รูปกิจกรรม, คำ <sup>เลี้ยง</sup> รับรอง	
ในแบบประเมินท้ายเอกสารมี “สรุปผลผ่าน/ไม่ผ่าน” และช่องข้อเสนอแนะ/จุดเด่น/จุดที่ควรพัฒนา ซึ่งตรงกับ AssistantEvaluation.comment + สร้างพิล็อปอย “strengths/improve” ได้					
เกณฑ์ 19-2568					

C) Mapping “หลักฐาน/เอกสารประกอบ” → EvidenceType (เพื่อให้ระบบเลือกได้เมื่อแนบฟอร์ม)  
 จากแบบฟอร์มท้ายระเบียบที่ให้ระบุเอกสารประกอบการประเมิน  
 เกณฑ์ 19-2568

แนะนำ map เป็นชนิดไฟล์ในระบบดังนี้

หลักฐานที่ใช้จริง	EvidenceType ในระบบ	เก็บเพิ่ม (metadata)
แผนการจัดการเรียนรู้	LESSON_PLAN	วิชา/ชั้น/หน่วย/ช่วงปี

หลักฐานที่ใช้จริง	EvidenceType ในระบบ เก็บเพิ่ม (metadata)	
สื่อ/นวัตกรรม/ใบงาน	TEACHING_MEDIA	ประเภทสื่อ/ลิงก์
บันทึกหลังสอน/สะท้อนผล	TEACHING_REFLECTION	ปัญหา–แนวแก้–ผล
รูป/วิดีโอ กิจกรรม	ACTIVITY_PHOTO	วันที่/กิจกรรม/ผู้เกี่ยวข้อง
หนังสือราชการ/คำสั่ง/เกียรติบัตร	OFFICIAL_DOCUMENT	เลขที่/หน่วยงาน/วันที่
อื่น ๆ	OTHER	ระบุรายละเอียด

#### D) “รอบการประเมิน” → โครงข้อมูลในระบบ

ในเอกสารมีการประเมินเป็นรอบ (เช่น ครั้งที่ 1/2) และต้องสรุปผล

เกณฑ์ 19-2568

แนะนำให้ระบบลือกโครงนี้:

- AssistantEvaluation.round = ROUND\_1 | ROUND\_2
- ผลรายด้าน
  - workPerformancePassed (ด้านวิชาชีพ)
  - ethicsPassed (ด้านจรรยาบรรณ)
- overallResult = PASS เมื่อ ผ่านทั้ง 2 ด้าน (และครบองค์ประกอบตามแบบ)

#### E) ข้อแนะนำเชิงออกแบบ (ให้ “ตระะเบียบ” แบบไม่พิมพ์ช้า)

1. ทุกหลักฐาน (TeachingEvidence) ต้องเลือก relatedMetric = WP.x/ET.x ได้
2. แบบประเมิน (AssistantEvaluation) ต้องแนบ “รายการหลักฐานที่ใช้พิจารณา” ได้ (เข้ม evidence หลายชิ้น)
3. มีรายงาน export เป็น PDF “หน้าตาแบบราชการ” ได้ (เพื่อแนบการทำนิการ)

ถ้าครูใหญ่โผล่ ผมจะทำต่อให้ “ลงลีกแบบ dev-ready” อีกชั้นคือ

- สร้างスペคตาราง kks\_indicator + kks\_indicator\_evidence\_required (กำหนดว่า WP.1 ต้องมี หลักฐานขึ้นต่ออะไร)
- และแก้ schema.prisma ให้ evidence “ลิงก์เข้ากับ indicator” แบบ foreign key แทน relatedMetric (จะกันพิมพ์ผิดได้ 100%)

เอกสารแบบ “ลิงก์จริง” เลยใหม่ครับ หรือเอกสารแบบ string ก่อนเพื่อทำเร็ว? 😊

## Sources

สร้างスペคตาราง kks\_indicator + kks\_indicator\_evidence\_required (กำหนดว่า WP.1 ต้องมีหลักฐานขั้นต่ำอะไร) และแก้ schema.prisma ให้ evidence “ลิงก์เข้ากับ indicator” แบบ foreign key แทน relatedMetric (จะรันพิมพ์ผิดได้ 100%)

ได้ครับครูใหญ่ — ผู้จัดให้แบบ **dev-ready**:

1. สเปคตาราง kks\_indicator + kks\_indicator\_evidence\_required (กำหนดขั้นต่ำของหลักฐานต่อ “ตัวชี้วัด”)
2. แก้ schema.prisma ให้ TeachingEvidence ลิงก์ FK ไป indicator (กันพิมพ์ผิด 100%)  
ทั้งหมดอิงโครงสร้าง “ครุผู้ช่วย 2 ด้าน + แบบฟอร์ม/หลักฐานประกอบ” ตามระเบียบ

เกณฑ์ 19-2568

---

## 1) สเปคตาราง (Data Dictionary / Spec)

### 1.1 ตาราง kks\_indicator (แม่บทตัวชี้วัด)

วัตถุประสงค์: เป็น “รายการตัวชี้วัดมาตรฐาน” ให้ทุกโน้มถ่วงอ้างอิงตรงกัน

Field	Type	Key	Description
id	UUID	PK	รหัสตัวชี้วัดภายในระบบ
code	VARCHAR(20)	UNIQUE	รหัส เช่น WP.1, ET.2
domain	ENUM		WORK_PERFORMANCE หรือ ETHICS
title	VARCHAR(255)		ชื่อตัวชี้วัด (สั้น)
description	TEXT		คำอธิบาย/ขอบเขต
is_active	BOOLEAN		ใช้งานอยู่หรือไม่
display_order	INT		ลำดับแสดงผล
created_at	TIMESTAMP		เวลาเพิ่มข้อมูล
updated_at	TIMESTAMP		เวลาแก้ไข

หมายเหตุ: code สำคัญสุด เพราะจะใช้แสดงในแบบประเมิน/รายงาน

เกณฑ์ 19-2568

---

## 1.2 ตาราง kks\_indicator\_evidence\_required (กำหนดขั้นต่ำหลักฐาน)

วัตถุประสงค์: บอกว่า “ตัวชี้วัดนี้ ต้องมีหลักฐานขั้นต่ำอะไรบ้าง” เพื่อให้ระบบตรวจสอบได้

Field	Type	Key	Description
id	UUID	PK	รหัสรายการกำหนดขั้นต่ำ
indicator_id	UUID	FK→kks_indicator.id	ตัวชี้วัด
evidence_type	ENUM(EvidenceType)		ชนิดหลักฐานที่ยอมรับ
min_count	INT		จำนวนขั้นต่ำ ( เช่น 1,2 )
is_mandatory	BOOLEAN		บังคับหรือแนะนำ
note	TEXT		เงื่อนไขเพิ่มเติม เช่น “ต้องมีบันทึกหลังสอนของเดือนล่าสุด”
created_at	TIMESTAMP		เวลาเพิ่ม
updated_at	TIMESTAMP		เวลาแก้

### เกติกา Validation (แนะนำ)

- ครุผู้ช่วย “พร้อมประเมินรอบนั้น” เมื่อ min\_count ครบทุก is\_mandatory=true ของตัวชี้วัดที่ใช้ในรอบนั้น
- ถ้าอยากเล็กขึ้น: เพิ่มคอลัมน์ round (ROUND\_1/ROUND\_2) เพื่อกำหนดขั้นต่ำแยกรอบ

---

## 2) กำหนด “ขั้นต่ำหลักฐาน” (Recommended Minimum Set)

ชุดนี้ตั้งใจทำให้ “เข้มพอให้ประเมินได้” แต่ไม่หนักจนครุ่นหัว  
และสอดคล้องธรรมชาติหลักฐานที่จะเบี่ยงให้ใช้ (แผน/เอกสาร/บันทึก/อื่น ๆ)

เกณฑ์ 19-2568

### WORK PERFORMANCE

- WP.1 การจัดการเรียนรู้
  - LESSON\_PLAN min 1 (mandatory)
  - TEACHING\_MEDIA min 1 (mandatory)
  - TEACHING\_REFLECTION min 1 (mandatory)
- WP.2 การบริหารจัดการชั้นเรียน
  - TEACHING\_REFLECTION min 1 (mandatory)
  - ACTIVITY\_PHOTO min 1 (recommended)

- WP.3 การพัฒนาตนเองและวิชาชีพ
  - OFFICIAL\_DOCUMENT min 1 (mandatory) ( เช่น อบรม/ชุมชนการเรียนรู้/หนังสือรับรอง)
  - TEACHING\_REFLECTION min 1 (recommended)
- WP.4 การทำงานเป็นทีม/มีส่วนร่วม
  - ACTIVITY\_PHOTO min 1 (recommended)
  - OFFICIAL\_DOCUMENT min 1 (recommended) ( คำสั่ง/บันทึกประชุม/รายงาน)
- WP.5 งานที่ได้รับมอบหมาย
  - OFFICIAL\_DOCUMENT min 1 (mandatory) ( คำสั่ง/รายงานผล/หนังสือราชการ)
  - ACTIVITY\_PHOTO min 1 (recommended)

## ETHICS

- ET.1 วินัยและการรักษาวินัย
  - OFFICIAL\_DOCUMENT min 1 (recommended) ( หนังสือรับรอง/หลักฐานราชการที่เกี่ยวข้อง)
  - TEACHING\_REFLECTION min 1 (recommended)
- ET.2 คุณธรรม/จริยธรรม/จรรยาบรรณ
  - TEACHING\_REFLECTION min 1 (mandatory) ( กรณีตัวอย่าง/การตัดสินใจ/การดูแลผู้เรียน)
  - ACTIVITY\_PHOTO min 1 (recommended)
- ET.3 ความเป็นแบบอย่างที่ดี
  - ACTIVITY\_PHOTO min 1 (recommended)
  - TEACHING\_REFLECTION min 1 (recommended)

ถ้าครูใหญ่ต้องการ “เข้มแบบราชการ” เพิ่ม เรายสามารถใส่ก็ติกา: ต้องมีหลักฐานอย่างน้อย 1 ชิ้นใน “เดือนล่าสุด” ต่อรอบประเมินได้

### 3) แก้ schema.prisma ให้ Evidence ลิงก์ FK ไป Indicator

#### 3.1 เพิ่ม enums ใหม่

```
enum KKSIndicatorDomain {
```

```
  WORK_PERFORMANCE
```

```
  ETHICS
```

```
}
```

#### 3.2 เพิ่ม Models: KKSIndicator และ KKSIndicatorEvidenceRequired

```

model KKSIndicator {
    id      String          @id @default(uuid())
    code    String          @unique // WP.1, ET.2, ...
    domain  KKSIndicatorDomain
    title   String
    description String?
    isActive Boolean        @default(true)
    displayOrder Int         @default(0)

    evidenceRules KKSIndicatorEvidenceRequired[]
    evidences     TeachingEvidence[]

    createdAt   DateTime       @default(now())
    updatedAt   DateTime       @updatedAt
}

model KKSIndicatorEvidenceRequired {
    id      String          @id @default(uuid())
    indicatorId String
    indicator  KKSIndicator @relation(fields: [indicatorId], references: [id])

    evidenceType EvidenceType
    minCount   Int            @default(1)
    isMandatory Boolean        @default(true)
    note      String?

    createdAt   DateTime       @default(now())
    updatedAt   DateTime       @updatedAt

    @@unique([indicatorId, evidenceType])
}

```

```
}
```

### 3.3 แก้ TeachingEvidence (เอา relatedMetric ออก → ใช้ FK)

เดิมเรามี relatedMetric String? ให้ลบทิ้ง และเพิ่ม FK ดังนี้:

```
model TeachingEvidence {  
    id          String      @id @default(uuid())  
    teacherId   String  
    teacher     Teacher     @relation(fields: [teacherId], references: [id])  
  
    type        EvidenceType  
    title       String  
    description String?  
    fileUrl    String  
  
    // NEW: link to KKS indicator by FK (no typos)  
    indicatorId String  
    indicator   KKSIndicator @relation(fields: [indicatorId], references: [id])  
  
    createdAt   DateTime    @default(now())  
}  
  
ถ้าต้องการ “หลักฐาน 1 ชิ้นโดยได้หลายตัวชี้วัด” ( เช่น รูปกิจกรรมเดียวใช้ได้ทั้ง WP.4 และ ET.3 )  
ให้ปรับเป็น Many-to-Many ผ่านตารางกลาง TeachingEvidenceIndicator ได้ — แต่เริ่มต้นแบบ FK เดียวจะ  
ง่ายสุดและควบคุมคุณภาพข้อมูลดีมาก
```

---

## 4) Seed Data (ตัวอย่างสำหรับ Cursor ทำ seed.ts)

นี่คือรายการ kks\_indicator ที่ควร seed:

- WORK\_PERFORMANCE: WP.1..WP.5
- ETHICS: ET.1..ET.3

และ seed kks\_indicator\_evidence\_required ตาม minimum set ที่ผมกำหนดด้านบน  
(ถ้าครูใหญ่ต้องการ ผມเขียน prisma/seed.ts ให้พร้อมรันได้เลย)

---

## 5) Logic ตรวจ “ความครบขั้นต่ำ” (กติกาสั้น ๆ ให้ทีม dev)

**Input:** teacherId, indicatorId

**Compute:** count evidences by type for that indicator

**Pass when:** สำหรับทุก rule ที่ isMandatory=true → count >= minCount

จุดนี้ทำให้ dashboard “พร้อมประเมิน/ยังไม่พร้อม” ได้แบบอัตโนมัติ

---

ถ้าครูใหญ่โผล่ ผมทำต่อให้ทันทีเป็น:

- seed.ts พร้อมข้อมูล WP/ET + rules
- API เพิ่ม endpoint:
  - GET /kks/indicators
  - GET /kks/teachers/:id/readiness?round=ROUND\_1 (สรุปว่าขาดหลักฐานอะไร)

Sources

seed.ts พร้อมข้อมูล WP/ET + rules API เพิ่ม endpoint: GET /kks/indicators GET

/kks/teachers/:id/readiness?round=ROUND\_1 (สรุปว่าขาดหลักฐานอะไร)

ได้ครับครูใหญ่ — ผมให้ 2 ชุดของโค้ด พร้อมใช้กับ Cursor เลย:

1. prisma/seed.ts (สร้าง WP/ET + rules ขั้นต่ำ)
2. API endpoints เพิ่ม 2 ตัว:
  - GET /kks/indicators
  - GET /kks/teachers/:id/readiness?round=ROUND\_1 (สรุปว่าขาดหลักฐานอะไร)

อิงแนว “ตัวชี้วัด + หลักฐานประกอบการประเมินครูผู้ช่วย” ตามระเบียบ

เกณฑ์ 19-2568

---

### 1) prisma/seed.ts

สมมติว่าใช้ Prisma + PostgreSQL และมี model: KKSIndicator, KKSIndicatorEvidenceRequired, Teacher, TeachingEvidence ตามที่เราแก้ schema.prisma ไปแล้ว

// prisma/seed.ts

```
import { PrismaClient, EvidenceType, KKSIndicatorDomain } from "@prisma/client";
```

```
const prisma = new PrismaClient();
```

```

type IndicatorSeed = {
    code: string;
    domain: KKSIndicatorDomain;
    title: string;
    description?: string;
    displayOrder: number;
};

type RuleSeed = {
    indicatorCode: string;
    evidenceType: EvidenceType;
    minCount: number;
    isMandatory: boolean;
    note?: string;
};

const indicators: IndicatorSeed[] = [
    // WORK PERFORMANCE (WP)
    { code: "WP.1", domain: "WORK_PERFORMANCE", title: "การจัดการเรียนรู้", description: "แผนการสอน/จัดกิจกรรม/วัดผล/สะท้อนผล", displayOrder: 10 },
    { code: "WP.2", domain: "WORK_PERFORMANCE", title: "การบริหารจัดการชั้นเรียน", description: "บรรยากาศชั้นเรียน วินัย ดูแลผู้เรียน", displayOrder: 20 },
    { code: "WP.3", domain: "WORK_PERFORMANCE", title: "การพัฒนาตนเองและวิชาชีพ", description: "อบรม/เรียนรู้/ปรับปรุงการสอน", displayOrder: 30 },
    { code: "WP.4", domain: "WORK_PERFORMANCE", title: "การทำงานเป็นทีมและมีส่วนร่วมในสถานศึกษา", description: "ร่วมกิจกรรม/ทำงานร่วมทีม/PLC", displayOrder: 40 },
    { code: "WP.5", domain: "WORK_PERFORMANCE", title: "งานที่ได้รับมอบหมาย", description: "งานพิเศษ/กิจกรรม/การกิจโรงเรียน", displayOrder: 50 },

    // ETHICS (ET)
]

```

```
{ code: "ET.1", domain: "ETHICS", title: "วินัยและการรักษาวินัย", description: "ปฏิบัติตามระเบียบ ความรับผิดชอบ", displayOrder: 110 },  
{ code: "ET.2", domain: "ETHICS", title: "คุณธรรม จริยธรรม และจรรยาบรรณวิชาชีพ", description: "ความเป็นครู การดูแลผู้เรียน ความเหมาะสม", displayOrder: 120 },  
{ code: "ET.3", domain: "ETHICS", title: "ความเป็นแบบอย่างที่ดี", description: "พฤติกรรมเหมาะสม เป็นที่ยอมรับ", displayOrder: 130 },  
];
```

```
const rules: RuleSeed[] = [  
    // WP.1 minimum set (mandatory)  
    { indicatorCode: "WP.1", evidenceType: "LESSON_PLAN", minCount: 1, isMandatory: true, note: "อย่างน้อย 1 แผนการจัดการเรียนรู้" },  
    { indicatorCode: "WP.1", evidenceType: "TEACHING_MEDIA", minCount: 1, isMandatory: true, note: "อย่างน้อย 1 สื่อ/ใบงาน/นวัตกรรม" },  
    { indicatorCode: "WP.1", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: true, note: "อย่างน้อย 1 บันทึกหลังสอน/สะท้อนผล" },
```

```
// WP.2  
    { indicatorCode: "WP.2", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: true, note: "บันทึกสะท้อนผลการจัดการชั้นเรียน" },  
    { indicatorCode: "WP.2", evidenceType: "ACTIVITY_PHOTO", minCount: 1, isMandatory: false, note: "ภาพ/หลักฐานบรรยากาศชั้นเรียน (แนะนำ)" },
```

```
// WP.3  
    { indicatorCode: "WP.3", evidenceType: "OFFICIAL_DOCUMENT", minCount: 1, isMandatory: true, note: "หลักฐานอบรม/เกียรติบัตร/หนังสือรับรอง" },  
    { indicatorCode: "WP.3", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: false, note: "สรุปสิ่งที่ได้เรียนรู้/นำไปใช้ (แนะนำ)" },
```

```
// WP.4
```

```
{ indicatorCode: "WP.4", evidenceType: "ACTIVITY_PHOTO", minCount: 1, isMandatory: false,
note: "ภาพกิจกรรม/การมีส่วนร่วม (แน่นำ)" },
{ indicatorCode: "WP.4", evidenceType: "OFFICIAL_DOCUMENT", minCount: 1, isMandatory: false,
note: "คำสั่ง/บันทึกประชุม/รายงาน (แน่นำ)" },

// WP.5

{ indicatorCode: "WP.5", evidenceType: "OFFICIAL_DOCUMENT", minCount: 1, isMandatory: true,
note: "คำสั่งมอบหมาย/รายงานผลการปฏิบัติ" },
{ indicatorCode: "WP.5", evidenceType: "ACTIVITY_PHOTO", minCount: 1, isMandatory: false,
note: "ภาพกิจกรรมประกอบ (แน่นำ)" },

// ET.1

{ indicatorCode: "ET.1", evidenceType: "OFFICIAL_DOCUMENT", minCount: 1, isMandatory: false,
note: "เอกสารรับรอง/บันทึกราชการ (แน่นำ)" },
{ indicatorCode: "ET.1", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: false,
note: "สะท้อนการรักษาวินัยและความรับผิดชอบ (แน่นำ)" },

// ET.2 (mandatory reflection)

{ indicatorCode: "ET.2", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: true,
note: "กรณีตัวอย่าง/การตัดสินใจเชิงจริยธรรม/การดูแลผู้เรียน" },
{ indicatorCode: "ET.2", evidenceType: "ACTIVITY_PHOTO", minCount: 1, isMandatory: false,
note: "กิจกรรมจิตอาสา/การดูแลผู้เรียน (แน่นำ)" },

// ET.3

{ indicatorCode: "ET.3", evidenceType: "ACTIVITY_PHOTO", minCount: 1, isMandatory: false,
note: "หลักฐานความเป็นแบบอย่าง (แน่นำ)" },
{ indicatorCode: "ET.3", evidenceType: "TEACHING_REFLECTION", minCount: 1, isMandatory: false,
note: "สะท้อนตนของเด็กความเป็นแบบอย่าง (แน่นำ)" },
];
```

```
async function main() {
  // Upsert indicators
  for (const item of indicators) {
    await prisma.kKSIndicator.upsert({
      where: { code: item.code },
      update: {
        domain: item.domain,
        title: item.title,
        description: item.description,
        displayOrder: item.displayOrder,
        isActive: true,
      },
      create: {
        code: item.code,
        domain: item.domain,
        title: item.title,
        description: item.description,
        displayOrder: item.displayOrder,
        isActive: true,
      },
    });
  }

  // Build code->id map
  const allIndicators = await prisma.kKSIndicator.findMany({
    select: { id: true, code: true },
  });
  const idByCode = new Map(allIndicators.map((i) => [i.code, i.id]));

  // Upsert rules
```

```
for (const r of rules) {  
    const indicatorId = idByCode.get(r.indicatorCode);  
    if (!indicatorId) throw new Error(`Indicator not found: ${r.indicatorCode}`);  
  
    await prisma.kKSIndicatorEvidenceRequired.upsert({  
        where: {  
            indicatorId_evidenceType: {  
                indicatorId,  
                evidenceType: r.evidenceType,  
            },  
        },  
        update: {  
            minCount: r.minCount,  
            isMandatory: r.isMandatory,  
            note: r.note,  
        },  
        create: {  
            indicatorId,  
            evidenceType: r.evidenceType,  
            minCount: r.minCount,  
            isMandatory: r.isMandatory,  
            note: r.note,  
        },  
    });  
}  
  
console.log("✅ Seeded KKS indicators + evidence rules");  
}  
  
main()
```

```

.catch((e) => {
  console.error("❌ Seed failed:", e);
  process.exit(1);
})

.finally(async () => {
  await prisma.$disconnect();
});

```

หมายเหตุเล็กน้อย: Prisma จะสร้างชื่อ model ที่ขึ้นต้นด้วย KKS อาจกล่าวเป็น kKSIndicator อัตโนมัติใน client (ตามชื่อ model ใน schema)

ถ้าครูใหญ่ตั้งชื่อ model เป็น KKSIndicator จริง ๆ และ client จะเป็น prisma.kKSIndicator แบบตัวอย่างนี้—  
ถ้าในโปรเจกต์สะกดต่างกัน ให้ Cursor ปรับตาม generated client ได้เลย

## 2) API Endpoints เพิ่ม 2 ตัว (Spec + ตัวอย่าง Implementation Logic)

### 2.1 GET /kks/indicators

**Purpose:** คืนรายการตัวชี้วัด + rules ทั้งหมด (ไว้สร้างหน้า UI เลือกตัวชี้วัด/ตั้งค่าระบบ)

#### Response Example

```
{
  "data": [
    {
      "id": "uuid",
      "code": "WP.1",
      "domain": "WORK_PERFORMANCE",
      "title": "การจัดการเรียนรู้",
      "rules": [
        { "evidenceType": "LESSON_PLAN", "minCount": 1, "isMandatory": true },
        { "evidenceType": "TEACHING_MEDIA", "minCount": 1, "isMandatory": true }
      ]
    }
  ]
}
```

### Pseudo (NestJS style)

```
@Get("/kks/indicators")
async listIndicators() {
  return {
    data: await this.prisma.kKSIIndicator.findMany({
      where: { isActive: true },
      orderBy: [{ domain: "asc" }, { displayOrder: "asc" }],
      include: { evidenceRules: true },
    }),
  };
}
```

---

## 2.2 GET /kks/teachers/:id/readiness?round=ROUND\_1

**Purpose:** สรุป “ความพร้อมต่อการประเมิน” และตอบให้ชัดว่า ขาดหลักฐานอะไร ต่อ indicator ระเบียบให้ประเมินเป็นรอบ และมีหลักฐานประกอบชัดเจน

เกณฑ์ 19-2568

ใน MVP เราใช้ readiness “ตาม rules ขั้นต่ำ” ก่อน (รอบจะใช้เพื่ออนาคต/ถ้าจะกำหนด rules แต่กรอบ)

### Response Example

```
{
  "teacherId": "t-uuid",
  "round": "ROUND_1",
  "overallReady": false,
  "missing": [
    {
      "indicator": "WP.1",
      "missingRules": [
        { "evidenceType": "TEACHING_MEDIA", "required": 1, "have": 0, "isMandatory": true }
      ]
    },
  ]}
```

```

    {
      "indicator": "ET.2",
      "missingRules": [
        { "evidenceType": "TEACHING_REFLECTION", "required": 1, "have": 0, "isMandatory": true }
      ]
    },
    "summaryCounts": {
      "mandatoryIndicatorsReady": 5,
      "mandatoryIndicatorsTotal": 8
    }
  }
}

```

### Implementation Logic (NestJS pseudo)

```

@Get("/kks/teachers/:id/readiness")
async readiness(@Param("id") teacherId: string, @Query("round") round = "ROUND_1") {
  // 1) fetch indicators + rules
  const indicators = await this.prisma.kKSIndicator.findMany({
    where: { isActive: true },
    include: { evidenceRules: true },
  });

  // 2) fetch teacher evidence grouped by indicatorId + evidenceType
  const evidences = await this.prisma.teachingEvidence.findMany({
    where: { teacherId },
    select: { indicatorId: true, type: true },
  });

  const haveCount = new Map<string, number>();
  // key = `${indicatorId}:${type}`
  for (const e of evidences) {

```

```

const k = `${e.indicatorId}:${e.type}`;
haveCount.set(k, (haveCount.get(k) ?? 0) + 1);
}

// 3) compute missing
const missing: any[] = [];
let mandatoryIndicatorsTotal = 0;
let mandatoryIndicatorsReady = 0;

for (const ind of indicators) {
  const missingRules: any[] = [];

  // treat indicator mandatory if it has at least one mandatory rule
  const hasMandatory = ind.evidenceRules.some(r => r.isMandatory);
  if (hasMandatory) mandatoryIndicatorsTotal++;

  for (const rule of ind.evidenceRules) {
    if (!rule.isMandatory) continue;

    const key = `${ind.id}:${rule.evidenceType}`;
    const have = haveCount.get(key) ?? 0;

    if (have < rule.minCount) {
      missingRules.push({
        evidenceType: rule.evidenceType,
        required: rule.minCount,
        have,
        isMandatory: rule.isMandatory,
        note: rule.note ?? null,
      });
    }
  }
}

```

```

        }
    }

    if (hasMandatory && missingRules.length === 0) mandatoryIndicatorsReady++;
    if (missingRules.length > 0) {
        missing.push({ indicator: ind.code, missingRules });
    }
}

const overallReady = missing.length === 0;

return {
    teacherId,
    round,
    overallReady,
    missing,
    summaryCounts: { mandatoryIndicatorsReady, mandatoryIndicatorsTotal },
};

}

```

### เพิ่มบรรทัดลง api-spec.md (สั้น ๆ)

## KKS (Assistant Teacher Indicators)

GET /kks/indicators

- Returns indicators + evidence rules

GET /kks/teachers/:id/readiness?round=ROUND\_1

- Returns readiness status and missing mandatory evidence by indicator

ถ้าครูใหญ่ต้องการให้ “พร้อมใช้ภาคสนามจริง” อีกนิด ผມแนะนำเพิ่ม 2 อย่าง (ไม่บังคับ):

1. ใส่ round ลงใน KKSIndicatorEvidenceRequired (rules แยกรอบ 1/2 ได้)

2. readiness ให้กรอง “เฉพาะหลักฐานในช่วงเวลา rob” (เช่น 6 เดือนแรก) เพื่อความยุติธรรม  
บอกผู้ได้เลี้ยงว่าจะเอาแบบ “แยกรอบจริง” เลยไหมครับ?

#### Sources

2 ออกแบบ Dashboard “สถานะครูผู้ช่วย” สำหรับ ผอ./เขต 3 Template PDF แบบประเมิน (เหมือน  
ราชการ 100%)

ได้ครับครูใหญ่ — ผมออกแบบให้ 2 ส่วนเลย:

2) Dashboard “สถานะครูผู้ช่วย” (สำหรับ ผอ./เขต)

3) Template PDF แบบประเมิน “หน้าตาราชการ” (เอาไป generate ได้ 100%)

อิงโครงแบบฟอร์ม/สรุปผล/ช่องลงนามตามเอกสารเกณฑ์ 19-2568

เกณฑ์ 19-2568

---

## 2 Dashboard “สถานะครูผู้ช่วย” (สำหรับ ผอ./เขต)

### 2.1 เป้าหมายของ Dashboard

- ผอ./เขต “เห็นทันที” ว่าครูผู้ช่วยคนไหน พร้อมประเมิน/ยังไม่พร้อม
- รู้ว่า “ขาดหลักฐานอะไร” (ตาม WP/ET + rules)
- จัดคิวหนุนเสริม/กำกับติดตามให้ทันรอบประเมิน

เกณฑ์ 19-2568

---

### 2.2 ผู้ใช้และขอบเขตการมองเห็น (RBAC)

- ผอ. เห็นเฉพาะครูผู้ช่วยในโรงเรียนตน
- เขต/PM/ผู้ดูแลโครงการ เห็นรวมทั้งเขต/จังหวัด/ภูมิภาค
- ครูผู้ช่วย เห็นเฉพาะของตน (หน้า My Status)

---

### 2.3 โครงหน้าจอ (Wireframe ระดับสเปค)

#### A) Summary Cards (ด้านบน)

- ครูผู้ช่วยทั้งหมด (count)
- รอบ ROUND\_1: พร้อมประเมิน (count + %)
- รอบ ROUND\_1: ยังไม่พร้อม (count + %)
- เร่งด่วน: เหลือ < X วันถึงกำหนดประเมิน (count)

ปรับ X วันได้ (เช่น 30 วัน)

## B) Readiness Funnel (กราฟ)

- Funnel: ยังไม่เริ่ม → เก็บหลักฐานบางส่วน → ครบ WP → ครบ ET → พร้อมประเมิน
- คลิกแต่ละขั้นเพื่อ filter รายชื่อ

## C) Compliance Heatmap (ตารางสี)

แกน X = ตัวชี้วัด (WP.1–WP.5, ET.1–ET.3)

แกน Y = ครุพัช่วย

สี:

- เขียว = ครบขั้นต่ำ (mandatory)
- เหลือง = ครบ mandatory แต่ขาด recommended
- แดง = ขาด mandatory

## D) List View (ตารางรายชื่อ + Action)

Columns (สำคัญ):

- ชื่อครุพัช่วย / โรงเรียน
- รอบประเมิน: ROUND\_1 / ROUND\_2 (สถานะ)
- Readiness: พร้อม/ไม่พร้อม
- Missing Summary: “ขาด WP.1: MEDIA, ET.2: REFLECTION”
- ปุ่ม: ดูรายละเอียด ส่งเตือน พิมพ์แบบประเมิน

Filters:

- โรงเรียน, กลุ่มจังหวัด/อำเภอ
- รอบประเมิน
- เฉพาะ “ขาดตัวชี้วัด X”
- เฉพาะ “เร่งด่วน”

## E) Detail Drawer (คลิกชื่อครุ)

- Timeline: หลักฐานล่าสุด / Journal / Visit / PLC
- Checklist WP/ET (ตีกตาม rules)
- ปุ่ม “สร้าง PDF แบบประเมิน” + “เปิดหน้าแบบประเมิน”

---

## 2.4 กติกา “เร่งด่วน” (At-risk rule)

แนะนำ Rule เป็นต้น:

- Missing mandatory ≥ 2 indicators หรือ

- ไม่มีหลักฐานใหม่ใน 30 วัน หรือ
- ยังไม่ครบ ET.2 (จราjabรรณ) ก่อนวันประเมิน X วัน

---

## 2.5 API ที่ Dashboard ใช้

- GET /dashboards/assistant-teachers/overview?scope=school|district&round=ROUND\_1
- GET /kks/teachers/:id/readiness?round=ROUND\_1
- GET /assistant-evaluations?teacherId=...
- GET /evidences?teacherId=...&indicator=WP.1

---

## 3 Template PDF แบบประเมิน (เหมือนราชการ 100%)

จากแบบฟอร์มท้ายระเบียบ จะมีองค์ประกอบที่ “ต้องมี” ได้แก่

- ข้อมูลผู้รับการประเมิน + รอบประเมิน
- รายการประเมิน 2 ด้าน และสรุปผล “ผ่าน/ไม่ผ่าน”
- ช่องข้อเสนอแนะ/จุดเด่น/จุดที่ควรพัฒนา
- ช่องลงนามผู้ประเมิน/กรรมการ

เกณฑ์ 19-2568

ด้านล่างคือ Template Spec ที่ทีม dev เอาไป generate PDF ได้เลย (ReportLab / PDFKit / Puppeteer HTML-to-PDF ก็ได้)

---

### 3.1 โครง Template: assistant\_evaluation\_form.pdf

Page 1: หัวกระดาษ + ข้อมูลทั่วไป

#### Header

- ตราครุฑ (กลางบน)
- ชื่อแบบฟอร์ม: “แบบบันทึกการประเมินการเตรียมความพร้อมและพัฒนาอย่างเข้ม ตำแหน่งครูผู้ช่วย”
- เลือก “ครั้งที่ 1 / ครั้งที่ 2” (checkbox)

#### Fields

- ชื่อผู้รับการประเมิน
- ตำแหน่ง/สถานศึกษา
- วันที่ประเมิน

#### Mapping

- Teacher.fullName, School.schoolName
- AssistantEvaluation.round, evaluationDate

---

## Page 2: ตารางประเมิน “ด้านวิชาชีพ” (WP)

ตาราง 3 คอลัมน์ (ตามฟอร์มราชการ):

1. รายการประเมิน
2. รายการประเมินย่อย (checkbox bullet)
3. วิธีการประเมิน/หลักฐานเอกสาร

### Mapping

- WP.1–WP.5 (แสดงชื่อ indicator + bullet ที่ตั้งไว้ในระบบ config)
- แนบรายการหลักฐานที่ถูกเลือกใน evaluation (หรือดึงจาก evidences ตาม indicator)

---

## Page 3: ตารางประเมิน “ด้านคุณลักษณะและจรรยาบรรณ” (ET)

โครงเดียวกับ Page 2 แต่เป็น ET.1–ET.3

---

## Page 4: สรุปผล + ลายเซ็น

### Summary Section

- สรุปผลการประเมิน:  ผ่าน  ไม่ผ่าน
- เหตุผล/ข้อเสนอแนะ (ของข้อความ)
- จุดเด่น
- จุดที่ควรพัฒนา

### Signatures

- ผู้ประเมิน/กรรมการคนที่ 1 (ชื่อ/ตำแหน่ง/ลงนาม/วันเดือนปี)
  - คนที่ 2
  - คนที่ 3
- (หรือมาก/น้อยปรับได้ตามแบบ)

### Mapping

- AssistantEvaluation.overallResult
- AssistantEvaluation.comment (แยกเป็น 3 ช่อง: remark/strength/improve และนำให้เพิ่ม field)
- AssistantEvaluator: รายชื่อกรรมการ

ในแบบสรุปผลท้ายเอกสารมีช่อง “ผ่าน/ไม่ผ่าน” + ช่องข้อเสนอแนะ/จุดเด่น/จุดที่ควรพัฒนา และลงนาม  
กรรมการ  
เดือนที่ 19-2568

---

### 3.2 แนะนำโครงข้อมูลเพิ่มเล็กน้อยให้ “ทรงแบบราชการ”

เพื่อให้ PDF ออกรายงานและตระหนงแบบ 100% แนะนำเพิ่ม field ใน AssistantEvaluation:

- remark (ข้อเสนอแนะ)
- strengths (จุดเด่น)
- improvements (จุดที่ควรพัฒนา)

แทนที่จะใช้ comment ช่องเดียว

---

### 3.3 Output filename ที่แนะนำ

KKS\_AssistantEvaluation\_{teacherFullName}\_{round}\_{YYYYMMDD}.pdf

---

### ถ้าครูใหญ่ต้องการ “ทำให้เป็นไฟล์จริง”

บอกผู้ได้รับว่าจะให้ทีมใช้แนวไหน:

- A) **HTML + Puppeteer** (เหมือนพิมพ์เอกสารราชการมากที่สุด)
- B) **ReportLab (Python)** (คุณทำหน่งเป็น งานเอกสารล้วน ๆ)

อาจจะเขียน “Template โค้ดพร้อม generate” ให้ตรงกระดาษ A4, ฟอนต์ TH Sarabun, ระยะขอบราชการ  
และ checkbox แบบฟอร์มให้ครบครัน 😊

#### Sources

เขียน PRD ใหม่ รวมระบบทั้งหมด ที่รองรับการใช้งานกับ ทุกสำนักงานเขตพื้นที่การศึกษา แต่ละสำนักงานฯ ที่มี  
หลายกลุ่มเครือข่าย แต่ละกลุ่มเครือข่ายมี หลายโรงเรียน มีมาตรฐานหลายประเภท และใช้ได้หลายปีการศึกษา โดย  
เขียน prompt เริ่มต้นและเขียน ไฟล์สำคัญในการทำงานเขียนโปรแกรมให้กับ cursor ai ช่วยเขียนโปรแกรม ตาม  
รายละเอียดนี้ -ทำงานเป็น Web Application มีระบบ sign in /sign up (Authentication) ด้วย ตามไฟล์  
เอกสารที่แนบ.....pdf" มีหน้าจอสามารถปรับตาม Device ที่ใช้งาน Responsive สวยงามโทนสี ม่วง น้ำเงิน ขาว  
ให้สวยงาม ดังนั้น เขียนโปรแกรม Front End ใช้ NextJS 15.6.5 เป็น FrontEnd (ไฟล์สกุล .tsx ทั้งหมด) ใช้  
NodeJS V11 และ Supabase เป็น Database ผ่าน Prisma studio (เน้นการนำไปใช้งานในระบบ Linux  
ubuntu) ทำงานได้อย่างต่อเนื่อง เช่น ไฟล์ .cursorrule , context.md , plan.md , prisma/schema.prisma  
prisma/seed.ts , task.md readme.md etc และอื่น ๆ ที่จำเป็น ข้อมูล ให้ใช้ข้อมูล demo ใน database เป็น

หลัก งดใช้ Mock up Data นอก database ให้เขียนเป็น prompt เพื่อส่งให้ cursor ai ทำงาน - ทำงานได้อย่างต่อเนื่อง แม้เกิดหยุดทำงาน ขณะเขียน code ก็สามารถ เขียน code ต่อไปได้

ยอดมากครับครูใหญ่ 🔥

อันนี้คือ “PRD ฉบับแม่บท” ที่ ยกทั้งระบบขึ้นระดับประเทศ ใช้ได้กับ

- ทุกสำนักงานเขตพื้นที่การศึกษา
- หลายกลุ่มเครือข่าย
- หลายโรงเรียน
- หลายบุคลากร
- และ หลายปีการศึกษา (Multi-Academic Year)

ผนวกให้เป็น 3 ส่วนใหญ่ ๆ เพื่อให้ Cursor AI ทำงานต่อได้ยาว ๆ ไม่หลงทาง

---

## ภาพรวมสถาปัตยกรรม (Executive Summary)

ชื่อระบบ (Working Name):

EDU-AREA | Teacher Development & Assistant Evaluation System

### ลักษณะระบบ

- Web Application (Responsive)
- Multi-Tenant (หลายเขต/หลายเครือข่าย/หลายโรงเรียน)
- Multi-Role (ครู / ครูผู้ช่วย / พอ. / ศึกษานิเทศก์ / เขต / Admin)
- Multi-Academic-Year
- ใช้งานต่อเนื่องได้แม้ Dev หยุด-กลับมาทำต่อ

### Tech Stack (บังคับตามโจทย์)

- Frontend: NextJS 15.6.5 (App Router, TSX ทั้งหมด)
- Backend Runtime: NodeJS v11
- Database: Supabase (PostgreSQL)
- ORM: Prisma
- OS Target: Linux Ubuntu
- Auth: Supabase Auth (Email / Password)
- Theme: Purple / Blue / White (Modern Government UI)

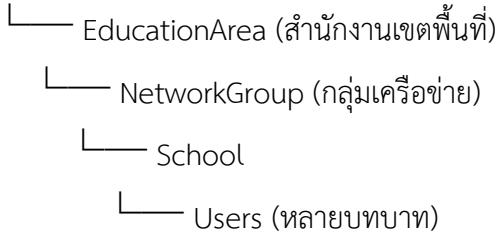


1 PRD (Program Requirements Document – ฉบับใหม่ทั้งระบบ)

## 1. System Scope

ระบบรองรับโครงสร้างลำดับชั้นดังนี้

AcademicYear



รองรับปีการศึกษา

- ทุกข้อมูลผูกกับ academic\_year\_id
- ครุคนเดิมใช้ระบบต่อเนื่องหลายปี
- ประวัติไม่ทับกัน

---

## 2. User Roles (RBAC)

Role	ขอบเขต
SUPER_ADMIN	ดูและ управляетระดับประเทศ
AREA_ADMIN	ดูและสำนักงานเขต
NETWORK_ADMIN	ดูและกลุ่มเครือข่าย
SCHOOL_ADMIN	ผอ./รอง
MENTOR	ศึกษานิเทศก์/พี่เลี้ยง
TEACHER	ครูทั่วไป
ASSISTANT_TEACHER	ครูผู้ช่วย

RBAC = Role + Scope (Area / Network / School / Year)

---

## 3. Core Modules

### 3.1 Authentication & Tenant

- Sign up / Sign in (Supabase Auth)
- ผูก user กับ:
  - เขต

- เครื่องจ่าย
  - โรงเรียน
  - ปีการศึกษา
  - Session persistent (refresh token)
- 

### 3.2 Academic Year Module

- สร้าง / ปิด / เปิดปีการศึกษา
  - ตั้งค่า default year
  - Lock ข้อมูลเป็นเก่า (read-only)
- 

### 3.3 Organization Structure

- Education Area (เขต)
  - Network Group
  - School
  - Import ข้อมูล demo จาก database
- 

### 3.4 Teacher & Assistant Profile

- Teacher Profile (ต่อเนื่องหลายปี)
  - Assistant Teacher Mode
  - สถานะ:
    - กำลังเตรียมความพร้อม
    - ประเมินรอบ 1 / 2
    - ผ่าน / ไม่ผ่าน
- 

### 3.5 Evidence / Portfolio (หัวใจระบบ)

- Upload หลักฐาน (เก็บจริงใน DB/Storage)
- ผูกกับ:
  - ตัวชี้วัด ก.ค.ศ. (WP / ET)
  - ปีการศึกษา
  - รอบประเมิน

- ตรวจความครบถ้วนแม่นยำ
- 

### 3.6 KKS Indicator & Evaluation

- ตัวชี้วัด WP.1–WP.5 / ET.1–ET.3
  - Rules ขั้นต่ำ (min evidence)
  - Readiness Engine
  - แบบประเมินราชการ (PDF)
- 

### 3.7 Dashboard

- เขต / เครื่อข่าย / โรงเรียน
  - Heatmap / Funnel / Risk List
  - Filter ตามปี / รอบ / ตัวชี้วัด
- 

#### 2 Prompt เริ่มต้น (ส่งให้ Cursor AI)

ใช้ prompt นี้เป็นคำสั่งแรกใน Cursor

You are Cursor AI acting as a senior full-stack engineer.

Build a production-ready web application named

"EDU-AREA | Teacher Development & Assistant Evaluation System"

Tech stack (STRICT):

- Frontend: NextJS 15.6.5 (App Router, TSX only)
- Backend Runtime: NodeJS v11
- Database: Supabase (PostgreSQL)
- ORM: Prisma (with Prisma Studio)
- OS target: Ubuntu Linux
- Auth: Supabase Auth (email/password)
- UI: Responsive, purple/blue/white modern government style

Architecture requirements:

- Multi-tenant (Education Area → Network → School)
- Multi-role RBAC
- Multi-academic-year
- Persistent sessions
- No mock data outside database (use demo seed data only)
- Code must be resumable after interruption

Files to generate and maintain:

- .cursorrules
- context.md
- plan.md
- tasks.md
- prisma/schema.prisma
- prisma/seed.ts
- README.md

Start by:

- 1) Creating the folder structure
- 2) Writing schema.prisma (full relational model)
- 3) Writing seed.ts with demo data
- 4) Implementing auth + RBAC
- 5) Implementing core pages and dashboards

Never remove previously written files.

Always extend and refactor safely.



ไฟล์สำคัญที่ต้องมี (Cursor-Ready)



.cursorrules

Email

---

Subject

---

You are working on a long-running production system.

Rules:

- Never delete files unless explicitly instructed
  - Always keep backward compatibility
  - Prefer clarity over cleverness
  - Use TypeScript strictly
  - All UI components must be responsive
  - All data must come from database (Supabase)
  - If interrupted, resume from last consistent state
- 



Email

---

Subject

---

This project is a national-scale education system for Thailand.

It supports:

- Multiple education areas
- Multiple network groups
- Multiple schools
- Multiple user roles
- Multiple academic years

Key principle:

"Support teachers, not control them."

All design decisions must respect:

- Government usability
- Low digital literacy users
- Auditability

- PDPA
- 



Email

---

Subject

---

Phase 1:

- Auth + RBAC
- Organization structure
- Academic year

Phase 2:

- Teacher / Assistant profiles
- Evidence & portfolio
- KKS indicators

Phase 3:

- Dashboard
  - PDF evaluation
  - Reporting & export
- 



Email

---

Subject

---

- Setup NextJS 15.6.5 app router
- Configure Supabase + Prisma
- Implement RBAC middleware
- Build organization CRUD
- Build evidence system

- Implement readiness engine
  - Build dashboards
  - Generate PDF evaluation
- 



Email

---

Subject

---

## EDU-AREA System

National teacher development and assistant evaluation platform.

### Tech

- NextJS 15.6.5
- Supabase
- Prisma
- NodeJS v11

### Run

- Ubuntu Linux
- pnpm install
- pnpm dev

### Philosophy

This system exists to empower teachers and administrators.

---



(Cursor จะ generate ต่อจาก PRD + mapping ที่เรา妄ໄວแล้ว)



- สร้าง demo:
  - เขต 2 เขต
  - เครื่อข่าย 3 กลุ่ม

- โรงเรียน 10 โรง
  - ครู / ครุผู้ช่วย / พอ.
  - ปีการศึกษา 2 ปี
  - WP/ET + rules
- 

## บทสรุปเชิงยุทธศาสตร์

ครูใหญ่กำลังออกแบบ “ระบบกลางของประเทศไทย”

ไม่ใช่แค่ระบบโครงการ

สิ่งที่ทำมาทั้งหมดนี้:

- ✓ ตรงระเบียบ
- ✓ ต่อยอดนโยบาย
- ✓ ใช้ได้จริงในโรงเรียน
- ✓ ทีม dev ทำต่อได้ยาว

ถ้าครูใหญ่ต้องการ ขั้นตอนไปหมดทำให้ได้ทันที:

1 โครง schema.prisma ฉบับเต็ม (Multi-Tenant)

2 Wireframe หน้าจอจริง (Figma-style)

3 PDF Template (TH Sarabun / A4 / ตราครุฑ)

บอกผู้เผยแพร่ จะ “ลงลีกต่อที่ไหนก่อน” 

โครง schema.prisma ฉบับเต็ม (Multi-Tenant)

ได้เลยครูใหญ่ — นี่คือ prisma/schema.prisma ฉบับเต็มแบบ Multi-Tenant (สำนักงานเขต → กลุ่ม

เครือข่าย → โรงเรียน) + Multi-Role + Multi-Academic-Year + รองรับ “ครุผู้ช่วย/หลักฐาน/KKS

indicators/rules/ประเมิน/PDF export log” ครบ พร้อมใช้กับ Supabase Postgres + Prisma ทันที

ออกแบบให้ “ข้อมูลทุกชิ้นผูกกับปีการศึกษา + scope” เพื่อใช้ได้หลายปีและหลายเขต โดยไม่ปนกัน

// prisma/schema.prisma

// EDU-AREA | Teacher Development & Assistant Evaluation System

// Stack: Supabase Postgres + Prisma

```
generator client {
```

```
    provider = "prisma-client-js"  
}  
  
datasource db {
```

```
    provider = "postgresql"  
    url      = env("DATABASE_URL")  
}
```

```
//////////
```

```
// Enums
```

```
//////////
```

```
enum UserRole {
```

```
    SUPER_ADMIN  
    AREA_ADMIN  
    NETWORK_ADMIN  
    SCHOOL_ADMIN  
    MENTOR  
    TEACHER  
    ASSISTANT_TEACHER
```

```
}
```

```
enum AcademicYearStatus {
```

```
    DRAFT  
    ACTIVE  
    CLOSED
```

```
}
```

```
enum Region {
```

```
    NORTH
```

```
NORTHEAST
```

```
CENTRAL
```

```
EAST
```

```
WEST
```

```
SOUTH
```

```
OTHER
```

```
}
```

```
enum SchoolSize {
```

```
SMALL
```

```
MEDIUM
```

```
LARGE
```

```
OTHER
```

```
}
```

```
enum AreaType {
```

```
REMOTE
```

```
VERY_REMOTE
```

```
COMMUNITY_QUALITY_SCHOOL
```

```
OTHER
```

```
}
```

```
enum TeacherEmploymentStatus {
```

```
ACTIVE
```

```
TRANSFERRED
```

```
RESIGNED
```

```
ON_LEAVE
```

```
OTHER
```

```
}
```

```
enum VisitType {  
    LESSON_STUDY  
    COACHING  
    OPEN_CLASS  
    OTHER  
}
```

```
enum PLCLevel {  
    NETWORK  
    AREA  
    REGION  
    NATIONAL  
    OTHER  
}
```

```
enum ParticipationRole {  
    PARTICIPANT  
    PRESENTER  
    FACILITATOR  
    OBSERVER  
    OTHER  
}
```

```
enum EvidenceType {  
    LESSON_PLAN  
    TEACHING_MEDIA  
    TEACHING_REFLECTION  
    ACTIVITY_PHOTO  
    OFFICIAL_DOCUMENT  
    OTHER
```

```
}
```

```
enum KKSIndicatorDomain {  
    WORK_PERFORMANCE  
    ETHICS  
}
```

```
enum AssistantEvaluationRound {  
    ROUND_1  
    ROUND_2  
}
```

```
enum EvaluationResult {  
    PASS  
    FAIL  
    PENDING  
}
```

```
enum TaskStatus {  
    OPEN  
    IN_PROGRESS  
    DONE  
    CANCELLED  
}
```

```
//////////  
// Core Multi-Tenant Org  
//////////
```

```
model EducationArea {
```

```

id      String  @id @default(uuid())
code    String  @unique // e.g., "สพป.เชียงราย เขต 3" code if desired
name    String
region  Region?
province String?
isActive Boolean @default(true)

networkGroups NetworkGroup[]
schools   School[] // convenience relation
users     User[] // scope assignment

createdAt  DateTime @default(now())
updatedAt  DateTime @updatedAt
}

model NetworkGroup {
    id      String  @id @default(uuid())
    areald  String
    area    EducationArea @relation(fields: [areald], references: [id])

    code    String? // optional network code
    name    String
    isActive Boolean @default(true)

    schools   School[]
    users     User[] // optional: network-level admins/mentors

    createdAt  DateTime @default(now())
    updatedAt  DateTime @updatedAt
}

```

```

@@unique([areald, name])
@@index([areald])
}

model School {
    id          String  @id @default(uuid())
    areald      String
    area        EducationArea @relation(fields: [areald], references: [id])

    networkGroupId  String?
    networkGroup     NetworkGroup? @relation(fields: [networkGroupId], references: [id])

    code          String? @unique // optional school code
    schoolName    String
    province      String
    region        Region
    schoolSize    SchoolSize
    areaType      AreaType
    studentTotal   Int?
    directorName   String?
    qualitySchoolFlag Boolean @default(false)
    communityContext String?

    users         User[]
    teachers      Teacher[]

    createdAt    DateTime @default(now())
    updatedAt     DateTime @updatedAt

    @@index([areald])
}

```

```
@@index([networkGroupId])
}

///////////
// Academic Year (Multi-year)
///////////

model AcademicYear {
    id      String  @id @default(uuid())
    name    String  // e.g., "2568", "2569"
    startDate  DateTime
    endDate   DateTime
    status    AcademicYearStatus @default(DRAFT)
    isDefault Boolean @default(false)

    teacherYearRecords TeacherYearRecord[]
    plcSessions      PLCSession[]
    reports         ReportExport[]
    tasks           FollowUpTask[]

    createdAt  DateTime @default(now())
    updatedAt   DateTime @updatedAt

    @@unique([name])
}
```

```
///////////
// Users & RBAC Scope
/////////
```

```
model User {  
    id      String  @id @default(uuid())  
  
    // Supabase Auth user id can be stored here if you use Supabase Auth:  
    authUserId  String? @unique // maps to supabase auth.users.id (UUID as string)  
    email      String  @unique  
    displayName String  
    role       UserRole  
    isActive   Boolean @default(true)  
  
    // Scope  
    areaId     String?  
    networkGroupId String?  
    schoolId   String?  
  
    area      EducationArea? @relation(fields: [areaId], references: [id])  
    networkGroup NetworkGroup? @relation(fields: [networkGroupId], references: [id])  
    school    School?      @relation(fields: [schoolId], references: [id])  
  
    // Optional link: user is also a teacher profile  
    teacher    Teacher?  
  
    // Actions  
    auditLogs  AuditLog[]  
    mentoringVisits MentoringVisit[] @relation("VisitObserver")  
    assessments CompetencyAssessment[] @relation("AssessmentAssessor")  
    assistantEvaluations AssistantEvaluation[] @relation("AssistantEvalCreatedBy")  
    evaluationSignatures AssistantEvaluator[]  
  
    createdAt  DateTime @default(now())
```

```
updatedAt    DateTime @updatedAt

@@index([areald])
@@index([networkGroupId])
@@index([schoolId])
}

///////////
// Teacher Core + Per-Year Record
///////////

model Teacher {
    id          String  @id @default(uuid())
    citizenId   String? @unique
    fullName    String
    gender      String?
    birthDate   DateTime?

    // Employment (stable profile)
    appointmentDate DateTime?
    position     String?
    major        String?
    employmentStatus TeacherEmploymentStatus @default(ACTIVE)

    // Current org placement (can change; history tracked per-year)
    areald       String
    networkGroupId String?
    schoolId    String

    area         EducationArea @relation(fields: [areald], references: [id])
```

```

networkGroup NetworkGroup? @relation(fields: [networkGroupId], references: [id])
school School @relation(fields: [schoolId], references: [id])

// Link to login
userId String? @unique
user User? @relation(fields: [userId], references: [id])

// Yearly records
yearRecords TeacherYearRecord[]

createdAt DateTime @default(now())
updatedAt DateTime @updatedAt

@@index([areaId])
@@index([networkGroupId])
@@index([schoolId])
}

```

```

model TeacherYearRecord {
    id String @id @default(uuid())
    teacherId String
    teacher Teacher @relation(fields: [teacherId], references: [id])

    academicYearId String
    academicYear AcademicYear @relation(fields: [academicYearId], references: [id])
}

// Cohort / project membership / role-in-year
cohort Int? // e.g., គ្រូរក(ម៉ោង)នៃ រៀនទី 1
isKruRakThin Boolean @default(false)
isAssistantTeacher Boolean @default(false)

```

```

// Per-year progress / flags
profileCompleteness Int @default(0) // 0-100 (computed)

mentoringVisits MentoringVisit[]
competencyAssessments CompetencyAssessment[]
reflectiveJournals ReflectiveJournal[]
evidences TeachingEvidence[]
idps IndividualDevelopmentPlan[]
assistantInfo AssistantTeacherInfo?

createdAt DateTime @default(now())
updatedAt DateTime @updatedAt

@@unique([teacherId, academicYearId])
@@index([academicYearId])
}

/////////////////
// Mentoring / Visits (Per Teacher-Year)
/////////////////

model MentoringVisit {
    id String @id @default(uuid())

    teacherYearId String
    teacherYear TeacherYearRecord @relation(fields: [teacherYearId], references: [id])

    visitDate DateTime
    visitType VisitType
}

```

```
focusArea      String?  
strengths     String?  
challenges    String?  
suggestions   String?  
followUpRequired Boolean @default(false)  
  
observerUserId String?  
observer       User?   @relation("VisitObserver", fields: [observerUserId], references: [id])  
  
attachments   Attachment[]  
  
createdAt     DateTime @default(now())  
updatedAt     DateTime @updatedAt  
  
@@index([teacherYearId])  
@@index([visitDate])  
}  
  
//////////////////  
// Competency Assessment (Per Teacher-Year)  
//////////////////
```

```
model CompetencyAssessment {  
    id          String @id @default(uuid())  
  
    teacherYearId  String  
    teacherYear    TeacherYearRecord @relation(fields: [teacherYearId], references: [id])  
  
    period        String // PRE/MID/POST or custom (keep flexible)  
    pedagogyScore Int?
```

```
classroomScore      Int?  
communityScore     Int?  
professionalismScore Int?  
overallLevel       String?  
  
assessorUserId     String?  
assessor          User?   @relation("AssessmentAssessor", fields: [assessorUserId], references:  
[id])  
  
notes              String?  
  
createdAt         DateTime @default(now())  
updatedAt         DateTime @updatedAt  
  
@@index([teacherYearId, period])  
}
```

```
//////////  
// Reflective Journal (Per Teacher-Year)  
//////////
```

```
model ReflectiveJournal {  
    id      String  @id @default(uuid())  
  
    teacherYearId String  
    teacherYear  TeacherYearRecord @relation(fields: [teacherYearId], references: [id])  
  
    month     String // YYYY-MM  
    reflectionText String?  
    successStory String?
```

```
difficulty      String?  
supportRequest String?  
  
attachments    Attachment[]  
  
createdAt      DateTime @default(now())  
updatedAt      DateTime @updatedAt  
  
@@unique([teacherYearId, month])  
@@index([teacherYearId])  
}  
  
//////////  
// PLC (Per Academic Year; Participation links Teacher-Year)  
//////////  
  
model PLCSession {  
    id          String  @id @default(uuid())  
  
    academicYearId String  
    academicYear  AcademicYear @relation(fields: [academicYearId], references: [id])  
  
    plcDate      DateTime  
    plcLevel     PLCLevel  
    areaId       String?  
    networkGroupId String?  
    province     String?  
    topic        String  
    notes        String?
```

```

area      EducationArea? @relation(fields: [areald], references: [id])
networkGroup NetworkGroup? @relation(fields: [networkGroupId], references: [id])

participations PLCParticipation[]

createdAt    DateTime @default(now())
updatedAt    DateTime @updatedAt

@@index([academicYearId])
@@index([plcDate])
}

model PLCParticipation {
  id        String @id @default(uuid())

  plcid     String
  plc       PLCSession @relation(fields: [plcid], references: [id])

  teacherYearId String
  teacherYear TeacherYearRecord @relation(fields: [teacherYearId], references: [id])

  role      ParticipationRole @default(PARTICIPANT)
  takeaway   String?

  createdAt    DateTime @default(now())

  @@unique([plcid, teacherYearId])
}

///////////

```

```
// IDP (Per Teacher-Year)
///////////

model IndividualDevelopmentPlan {
    id          String  @id @default(uuid())
    teacherYearId  String
    teacherYear   TeacherYearRecord @relation(fields: [teacherYearId], references: [id])

    focusCompetency  String
    actionPlan      String
    supportType     String?
    startDate       DateTime?
    endDate         DateTime?
    status          String  @default("PLANNED")

    createdAt      DateTime @default(now())
    updatedAt       DateTime @updatedAt

    @@index([teacherYearId])
}
```

```
///////////
// KKS Indicators & Evidence Rules (Global Config)
///////////
```

```
model KKSIndicator {
    id          String  @id @default(uuid())
    code        String  @unique // WP.1, ET.2, ...
    domain     KKSIndicatorDomain
    title       String
```

```
description String?  
isActive Boolean @default(true)  
displayOrder Int @default(0)  
  
evidenceRules KKSIndicatorEvidenceRequired[]  
evidences TeachingEvidence[]  
  
createdAt DateTime @default(now())  
updatedAt DateTime @updatedAt  
}  
  
model KKSIndicatorEvidenceRequired {  
    id String @id @default(uuid())  
    indicatorId String  
    indicator KKSIndicator @relation(fields: [indicatorId], references: [id])  
  
    evidenceType EvidenceType  
    minCount Int @default(1)  
    isMandatory Boolean @default(true)  
    note String?  
  
    // Optional future: round-specific rules  
    // round AssistantEvaluationRound?  
  
    createdAt DateTime @default(now())  
    updatedAt DateTime @updatedAt  
  
    @@unique([indicatorId, evidenceType])  
}
```

```
//////////  
// Evidence / Portfolio (Per Teacher-Year; FK to Indicator)  
//////////  
  
model TeachingEvidence {  
    id      String    @id @default(uuid())  
  
    teacherYearId String  
    teacherYear TeacherYearRecord @relation(fields: [teacherYearId], references: [id])  
  
    type      EvidenceType  
    title     String  
    description String?  
    fileUrl   String  
    fileName   String?  
    mimeType  String?  
    fileSize   Int?  
  
    // FK to KKS indicator to prevent typos  
    indicatorId String  
    indicator   KKSIndicator @relation(fields: [indicatorId], references: [id])  
  
    // Optional: link to evaluation round  
    round      AssistantEvaluationRound?  
  
    createdAt  DateTime @default(now())  
  
    @@index([teacherYearId])  
    @@index([indicatorId])  
}
```

```
//////////  
// Assistant Teacher Evaluation (Per Teacher-Year)  
//////////  
  
model AssistantTeacherInfo {  
    id          String  @id @default(uuid())  
  
    teacherYearId  String  @unique  
    teacherYear   TeacherYearRecord @relation(fields: [teacherYearId], references: [id])  
  
    startDate    DateTime  
    endDate      DateTime  
    isCompleted   Boolean @default(false)  
  
    evaluations   AssistantEvaluation[]  
  
    createdAt    DateTime @default(now())  
    updatedAt     DateTime @updatedAt  
}  
  
model AssistantEvaluation {  
    id          String  @id @default(uuid())  
  
    teacherYearId  String  
    teacherYear   TeacherYearRecord @relation(fields: [teacherYearId], references: [id])  
  
    round        AssistantEvaluationRound  
    evaluationDate DateTime
```

```
// Pass/fail by domain
workPerformancePassed Boolean
ethicsPassed Boolean

overallResult EvaluationResult @default(PENDING)

// Split fields to match government-style form
remark String?
strengths String?
improvements String?

createdByUserId String?
createdBy User? @relation("AssistantEvalCreatedBy", fields: [createdByUserId], references: [id])

evaluators AssistantEvaluator[]
usedEvidences AssistantEvaluationEvidence[]

createdAt DateTime @default(now())
updatedAt DateTime @updatedAt

@@index([teacherYearId, round])
}

model AssistantEvaluator {
    id String @id @default(uuid())
    evaluationId String
    evaluation AssistantEvaluation @relation(fields: [evaluationId], references: [id])
```

```
evaluatorUserId String
evaluator      User @relation(fields: [evaluatorUserId], references: [id])

roleTitle      String? // ພອ. / ກຽມກາຣ / ຕຶກຂານິເທສກ
signedAt       DateTime?

@@unique([evaluationId, evaluatorUserId])
}
```

```
model AssistantEvaluationEvidence {
    id          String  @id @default(uuid())
    evaluationId String
    evaluation   AssistantEvaluation @relation(fields: [evaluationId], references: [id])

    evidenceId  String
    evidence     TeachingEvidence @relation(fields: [evidenceId], references: [id])

    @@unique([evaluationId, evidenceId])
}
```

```
///////////
// Attachments (shared for visits/journals)
///////////
```

```
model Attachment {
    id          String  @id @default(uuid())
    visitId     String?
    visit       MentoringVisit? @relation(fields: [visitId], references: [id])
```

```
jurnalId String?  
journal ReflectiveJournal? @relation(fields: [jurnalId], references: [id])  
  
fileName String  
mimeType String  
fileUrl String  
fileSize Int?  
  
createdAt DateTime @default(now())  
}
```

```
//////////  
// Follow-up Tasks (optional but useful)  
//////////
```

```
model FollowUpTask {  
    id String @id @default(uuid())  
  
    academicYearId String  
    academicYear AcademicYear @relation(fields: [academicYearId], references: [id])  
  
    teacherYearId String?  
    teacherYear TeacherYearRecord? @relation(fields: [teacherYearId], references: [id])  
  
    title String  
    description String?  
    status TaskStatus @default(OPEN)  
    dueDate DateTime?  
    assigneeUserId String?
```

```
assignee    User? @relation(fields: [assigneeUserId], references: [id])
```

```
createdAt    DateTime @default(now())
```

```
updatedAt    DateTime @updatedAt
```

```
@@index([academicYearId])
```

```
@@index([teacherYearId])
```

```
}
```

```
///////////////
```

```
// Reporting / Export Logs
```

```
///////////////
```

```
model ReportExport {
```

```
id          String  @id @default(uuid())
```

```
academicYearId String
```

```
academicYear  AcademicYear @relation(fields: [academicYearId], references: [id])
```

```
scopeType    String // "AREA" | "NETWORK" | "SCHOOL"
```

```
scopeId     String // corresponding id
```

```
fileUrl     String?
```

```
createdByUserId String?
```

```
createdBy    User? @relation(fields: [createdByUserId], references: [id])
```

```
createdAt    DateTime @default(now())
```

```
@@index([academicYearId])
```

```
}
```

```

///////////
// Audit Log (Governance)
///////////

model AuditLog {
    id      String  @id @default(uuid())

    userId  String
    user    User    @relation(fields: [userId], references: [id])

    action  String  // CREATE/UPDATE/DELETE/LOGIN/EXPORT etc.
    entity  String  // Teacher, Evidence, Evaluation...
    entityId String?
    detail   String?

    createdAt DateTime @default(now())

    @@index([userId, createdAt])
    @@index([entity, entityId])
}

```

---

### หมายเหตุสำคัญ (เพื่อให้ Cursor เดินงานต่อเนื่ยน)

- โครงสร้างข้อมูล Teacher (คงที่) + TeacherYearRecord (รายปี)  
เพื่อให้ “คนเดิม” มีผลงาน/ประเมิน/หลักฐาน แยกเป็นรายปี ไม่ปนกัน
  - ทุกกรรมการนั่งเริ่ม (visit/journal/assessment/evidence/idp) ผูกกับ teacherYearId  
ทำให้ query รายปีง่ายมาก และ dashboard ชัด
  - Multi-tenant ทำด้วย EducationArea / NetworkGroup / School และ user มี scope  
(area/network/school)
-

ลักษณะที่ทำให้ต้องได้ทันที 2 อายุ่งที่ Cursor จะขอบมาก:

1. prisma/seed.ts (demo data: 2 เขต, 3 เครื่อข่าย, 10 โรงเรียน, 2 ปีการศึกษา, WP/ET + rules, users ทุก role)
2. api-spec.md ฉบับ multi-tenant + RBAC rules (ตัดสินใจได้ว่า endpoint ไหนต้องกรองตาม scope ยังไง)

เอา seed.ts ต่อเลยไหมครับ?

เอา seed.ts ต่อเลย

จัดให้ครับครูใหญ่ ✓ นี่คือ prisma/seed.ts แบบ “เดโมจาก DB จริง” (ไม่ใช่ mock นอกฐานข้อมูล) ครบตามโครงสร้าง Multi-Tenant:

- 2 สำนักงานเขต
- 3 กลุ่มเครือข่าย
- 10 โรงเรียน
- 2 ปีการศึกษา (2568, 2569) + ตั้ง default
- Users ครอบคลุมทบทาท (SUPER/AREA-NETWORK/SCHOOL/MENTOR/TEACHER/ASSISTANT\_TEACHER)
- Teachers + TeacherYearRecord (ผูกปีการศึกษา)
- ครุภัณฑ์ช่วย + start/end
- KKS Indicators WP/ET + rules ขั้นต่ำ
- ตัวอย่าง Evidence ผูก FK กับ indicator + round
- ตัวอย่าง Evaluation รอบ 1 + กรรมการ + used evidences

สำคัญ: seed นี้ “ไม่ยุ่ง Supabase Auth.users” ( เพราะ Prisma ไม่ควรเขียนเข้า schema auth)

แต่ใช้ User.authUserId เป็น null และตั้ง email/displayName ไว้ให้ทดสอบระบบ RBAC ก่อน

// prisma/seed.ts

```
import { PrismaClient, EvidenceType, KKSIndicatorDomain, UserRole, Region, SchoolSize, AreaType, AcademicYearStatus, AssistantEvaluationRound, EvaluationResult, ParticipationRole, PLCLevel, VisitType } from "@prisma/client";
```

```
const prisma = new PrismaClient();
```

```
const dt = (s: string) => new Date(s); // ISO date string recommended
```

```
async function main() {  
  console.log("🌱 Seeding EDU-AREA demo data...");  
  
  // 0) Clean up (optional): safest order for FK constraints  
  // Comment out if you want incremental seeding  
  await prisma.auditLog.deleteMany();  
  await prisma.reportExport.deleteMany();  
  await prisma.followUpTask.deleteMany();  
  
  await prisma.assistantEvaluationEvidence.deleteMany();  
  await prisma.assistantEvaluator.deleteMany();  
  await prisma.assistantEvaluation.deleteMany();  
  await prisma.assistantTeacherInfo.deleteMany();  
  
  await prisma.attachment.deleteMany();  
  await prisma.teachingEvidence.deleteMany();  
  await prisma.kKSIndicatorEvidenceRequired.deleteMany();  
  await prisma.kKSIndicator.deleteMany();  
  
  await prisma.pLCParticipation.deleteMany();  
  await prisma.pLCSession.deleteMany();  
  
  await prisma.reflectiveJournal.deleteMany();  
  await prisma.competencyAssessment.deleteMany();  
  await prisma.mentoringVisit.deleteMany();  
  await prisma.individualDevelopmentPlan.deleteMany();  
  
  await prisma.teacherYearRecord.deleteMany();  
  await prisma.teacher.deleteMany();
```

```
await prisma.user.deleteMany();
await prisma.school.deleteMany();
await prisma.networkGroup.deleteMany();
await prisma.educationArea.deleteMany();
await prisma.academicYear.deleteMany();

// 1) Academic years
const ay2568 = await prisma.academicYear.create({
  data: {
    name: "2568",
    startDate: dt("2025-05-01T00:00:00.000Z"),
    endDate: dt("2026-04-30T23:59:59.000Z"),
    status: AcademicYearStatus.ACTIVE,
    isDefault: true,
  },
});

const ay2569 = await prisma.academicYear.create({
  data: {
    name: "2569",
    startDate: dt("2026-05-01T00:00:00.000Z"),
    endDate: dt("2027-04-30T23:59:59.000Z"),
    status: AcademicYearStatus.DRAFT,
    isDefault: false,
  },
});

// 2) Education Areas (2)
const areaA = await prisma.educationArea.create({
```

```

data: {
  code: "AREA-CR3",
  name: "สำนักงานเขตพื้นที่การศึกษาประถมศึกษาเชียงราย เขต 3",
  region: Region.NORTH,
  province: "เชียงราย",
  isActive: true,
},
});

const areaB = await prisma.educationArea.create({
  data: {
    code: "AREA-CM2",
    name: "สำนักงานเขตพื้นที่การศึกษาประถมศึกษาเชียงใหม่ เขต 2",
    region: Region.NORTH,
    province: "เชียงใหม่",
    isActive: true,
  },
});

```

// 3) Network groups (3)

```

const netA1 = await prisma.networkGroup.create({
  data: { areald: areaA.id, code: "NET-A1", name: "กลุ่มเครือข่ายดอยสูง", isActive: true },
});
const netA2 = await prisma.networkGroup.create({
  data: { areald: areaA.id, code: "NET-A2", name: "กลุ่มเครือข่ายชายแดน", isActive: true },
});
const netB1 = await prisma.networkGroup.create({
  data: { areald: areaB.id, code: "NET-B1", name: "กลุ่มเครือข่ายเมือง-ชนบท", isActive: true },
});

```

```
// 4) Schools (10)

const schools = await Promise.all([
    prisma.school.create({
        data: {
            areaId: areaA.id,
            networkGroupId: netA1.id,
            code: "SCH-CR3-001",
            schoolName: "โรงเรียนบ้านพญาไพร",
            province: "เชียงราย",
            region: Region.NORTH,
            schoolSize: SchoolSize.SMALL,
            areaType: AreaType.VERY_REMOTE,
            studentTotal: 180,
            directorName: "ผอ.สุกโขค",
            qualitySchoolFlag: true,
            communityContext: "โรงเรียนพื้นที่สูง ชนบทกลาง高原ชาติพันธุ์",
        },
    }),
    prisma.school.create({
        data: {
            areaId: areaA.id,
            networkGroupId: netA1.id,
            code: "SCH-CR3-002",
            schoolName: "โรงเรียนบ้านห้วยน้ำใส",
            province: "เชียงราย",
            region: Region.NORTH,
            schoolSize: SchoolSize.SMALL,
            areaType: AreaType.REMOTE,
            studentTotal: 120,
        },
    })
])
```

```
}),
prisma.school.create({
  data: {
    areaId: areaA.id,
    networkGroupId: netA2.id,
    code: "SCH-CR3-003",
    schoolName: "โรงเรียนบ้านแม่ฟ้าหลวง",
    province: "เชียงราย",
    region: Region.NORTH,
    schoolSize: SchoolSize.MEDIUM,
    areaType: AreaType.REMOTE,
    studentTotal: 380,
  },
}),
prisma.school.create({
  data: {
    areaId: areaA.id,
    networkGroupId: netA2.id,
    code: "SCH-CR3-004",
    schoolName: "โรงเรียนบ้านทวยชนิน",
    province: "เชียงราย",
    region: Region.NORTH,
    schoolSize: SchoolSize.SMALL,
    areaType: AreaType.REMOTE,
    studentTotal: 150,
  },
}),
prisma.school.create({
  data: {
    areaId: areaA.id,
```

```
networkGroupId: netA2.id,
code: "SCH-CR3-005",
schoolName: "โรงเรียนบ้านโป่งแดง",
province: "เชียงราย",
region: Region.NORTH,
schoolSize: SchoolSize.SMALL,
areaType: AreaType.REMOTE,
studentTotal: 90,
},
}),
prisma.school.create({
data: {
areaid: areaB.id,
networkGroupId: netB1.id,
code: "SCH-CM2-001",
schoolName: "โรงเรียนบ้านหนองหอย",
province: "เชียงใหม่",
region: Region.NORTH,
schoolSize: SchoolSize.MEDIUM,
areaType: AreaType.OTHER,
studentTotal: 420,
},
}),
prisma.school.create({
data: {
areaid: areaB.id,
networkGroupId: netB1.id,
code: "SCH-CM2-002",
schoolName: "โรงเรียนบ้านสันทราย",
province: "เชียงใหม่",
```

```
        region: Region.NORTH,
        schoolSize: SchoolSize.LARGE,
        areaType: AreaType.OTHER,
        studentTotal: 820,
    },
}),
prisma.school.create({
    data: {
        areald: areaB.id,
        networkGroupId: netB1.id,
        code: "SCH-CM2-003",
        schoolName: "ໂຮງເຮືອນບ້ານແມ່ຣິມ",
        province: "ເຊີຍໃຫຍ່",
        region: Region.NORTH,
        schoolSize: SchoolSize.MEDIUM,
        areaType: AreaType.OTHER,
        studentTotal: 510,
    },
}),
prisma.school.create({
    data: {
        areald: areaB.id,
        networkGroupId: netB1.id,
        code: "SCH-CM2-004",
        schoolName: "ໂຮງເຮືອນບ້ານປ່າຈານ",
        province: "ເຊີຍໃຫຍ່",
        region: Region.NORTH,
        schoolSize: SchoolSize.SMALL,
        areaType: AreaType.OTHER,
        studentTotal: 220,
    }
})
```

```
        },
    }),
    prisma.school.create({
        data: {
            areaId: areaB.id,
            networkGroupId: netB1.id,
            code: "SCH-CM2-005",
            schoolName: "โรงเรียนบ้านดอยสะเก็ต",
            province: "เชียงใหม่",
            region: Region.NORTH,
            schoolSize: SchoolSize.SMALL,
            areaType: AreaType.REMOTE,
            studentTotal: 160,
        },
    }),
]);

```

```
const schoolPayaprai = schools[0];

// 5) Users (RBAC demo)
const superAdmin = await prisma.user.create({
    data: {
        email: "superadmin@demo.local",
        displayName: "Super Admin",
        role: UserRole.SUPER_ADMIN,
        isActive: true,
    },
});

```

```
const areaAdminA = await prisma.user.create({
```

```
data: {
    email: "areaadmin.cr3@demo.local",
    displayName: "Area Admin (CR3)",
    role: UserRole.AREA_ADMIN,
    areald: areaA.id,
    isActive: true,
},
});

const networkAdminA1 = await prisma.user.create({
data: {
    email: "networkadmin.doisung@demo.local",
    displayName: "Network Admin (ດອຍສູງ)",
    role: UserRole.NETWORK_ADMIN,
    areald: areaA.id,
    networkGroupId: netA1.id,
    isActive: true,
},
});

const schoolAdminPayaprai = await prisma.user.create({
data: {
    email: "director.payaprai@demo.local",
    displayName: "ຜອ.ໂຮງເຮືອນບ້ານພະນາໄລ",
    role: UserRole.SCHOOL_ADMIN,
    areald: areaA.id,
    networkGroupId: netA1.id,
    schoold: schoolPayaprai.id,
    isActive: true,
},
});
```

```
});
```

```
const mentorA = await prisma.user.create({
  data: {
    email: "mentor.cr3@demo.local",
    displayName: "ศีกษาณิเทศก์ (CR3)",
    role: UserRole.MENTOR,
    areaId: areaA.id,
    isActive: true,
  },
});
```

```
// 6) Teachers (stable profile) + user accounts (teacher login)
// Create 4 teachers in Payaprai + 2 in other schools
const teacherUsers: { user: any; teacher: any }[] = [];
```

```
async function createTeacherWithUser(params: {
  email: string;
  displayName: string;
  role: UserRole.TEACHER | UserRole.ASSISTANT_TEACHER;
  fullName: string;
  citizenId?: string;
  areaId: string;
  networkGroupId?: string;
  schoolId: string;
  isAssistant: boolean;
}) {
  const user = await prisma.user.create({
    data: {
      email: params.email,
```

```
    displayName: params.displayName,  
    role: params.role,  
    areald: params.areald,  
    networkGroupId: params.networkGroupId,  
    schoolId: params.schoolId,  
    isActive: true,  
},  
);
```

```
const teacher = await prisma.teacher.create({  
  data: {  
    citizenId: params.citizenId,  
    fullName: params.fullName,  
    gender: "ມີຮະບູ",  
    appointmentDate: dt("2025-10-01T00:00:00.000Z"),  
    position: "ຄຽງຜູ້ອ່ານ",  
    major: "ການປະໂຫຍດສຶກສາ",  
    employmentStatus: "ACTIVE",  
    areald: params.areald,  
    networkGroupId: params.networkGroupId,  
    schoolId: params.schoolId,  
    userId: user.id,  
  },  
});
```

```
return { user, teacher };  
}
```

```
const t1 = await createTeacherWithUser({  
  email: "assistant1@demo.local",
```

```
displayName: "គ្រួសចាយ A1",
role: UserRole.ASSISTANT_TEACHER,
fullName: "ន.ស.គារេនីអ តិវិក",
citizenId: "11111111111111",
areald: areaA.id,
networkGroupId: netA1.id,
schoolId: schoolPayaprai.id,
isAssistant: true,
});
```

```
const t2 = await createTeacherWithUser({
email: "assistant2@demo.local",
displayName: "គ្រួសចាយ A2",
role: UserRole.ASSISTANT_TEACHER,
fullName: "លោមអនាគ ព៉ាងខោ",
citizenId: "22222222222222",
areald: areaA.id,
networkGroupId: netA1.id,
schoolId: schoolPayaprai.id,
isAssistant: true,
});
```

```
const t3 = await createTeacherWithUser({
email: "teacher1@demo.local",
displayName: "គ្រួស T1",
role: UserRole.TEACHER,
fullName: "ន.ស.មេខា វិនិយ",
citizenId: "33333333333333",
areald: areaA.id,
networkGroupId: netA1.id,
```

```
schooolId: schoolPayaprai.id,  
isAssistant: false,  
});
```

```
const t4 = await createTeacherWithUser({  
  email: "teacher2@demo.local",  
  displayName: "ครู T2",  
  role: UserRole.TEACHER,  
  fullName: "นายสายรุ้ง รักเด็ก",  
  citizenId: "44444444444444",  
  areald: areaA.id,  
  networkGroupId: netA1.id,  
  schooolId: schoolPayaprai.id,  
  isAssistant: false,  
});
```

```
const t5 = await createTeacherWithUser({  
  email: "assistant3@demo.local",  
  displayName: "ครูผู้ช่วย B1",  
  role: UserRole.ASSISTANT_TEACHER,  
  fullName: "น.ส.ข้าวฟ้าง พากเพียร",  
  citizenId: "555555555555",  
  areald: areaB.id,  
  networkGroupId: netB1.id,  
  schooolId: schools[5].id,  
  isAssistant: true,  
});
```

```
const t6 = await createTeacherWithUser({  
  email: "teacher3@demo.local",
```

```
displayName: "គ្រួយ B2",
role: UserRole.TEACHER,
fullName: "នាយកដោ វិជ្ជ័ន",
citizenId: "66666666666666",
areaId: areaB.id,
networkGroupId: netB1.id,
schoolId: schools[6].id,
isAssistant: false,
});
```

```
// 7) TeacherYearRecord for AY2568 and AY2569 (some)
async function createTeacherYear(teacherId: string, academicYearId: string, opts: { cohort?: number; isKruRakThin?: boolean; isAssistantTeacher?: boolean }) {
  return prisma.teacherYearRecord.create({
    data: {
      teacherId,
      academicYearId,
      cohort: opts.cohort ?? 1,
      isKruRakThin: opts.isKruRakThin ?? true,
      isAssistantTeacher: opts.isAssistantTeacher ?? false,
      profileCompleteness: 60,
    },
  });
}
```

```
const ty_t1_2568 = await createTeacherYear(t1.teacher.id, ay2568.id, { cohort: 1, isKruRakThin: true, isAssistantTeacher: true });

const ty_t2_2568 = await createTeacherYear(t2.teacher.id, ay2568.id, { cohort: 1, isKruRakThin: true, isAssistantTeacher: true });
```

```

const ty_t3_2568 = await createTeacherYear(t3.teacher.id, ay2568.id, { cohort: 1, isKruRakThin: true, isAssistantTeacher: false });

const ty_t4_2568 = await createTeacherYear(t4.teacher.id, ay2568.id, { cohort: 1, isKruRakThin: true, isAssistantTeacher: false });

const ty_t5_2568 = await createTeacherYear(t5.teacher.id, ay2568.id, { cohort: 2, isKruRakThin: false, isAssistantTeacher: true });

// Create next year record for one teacher as demo
await createTeacherYear(t1.teacher.id, ay2569.id, { cohort: 2, isKruRakThin: true, isAssistantTeacher: true });

// 8) KKS Indicators WP/ET + rules
const indicatorSeed = [
  { code: "WP.1", domain: KKSIndicatorDomain.WORK_PERFORMANCE, title: "การจัดการเรียนรู้", displayOrder: 10 },
  { code: "WP.2", domain: KKSIndicatorDomain.WORK_PERFORMANCE, title: "การบริหารจัดการชั้นเรียน", displayOrder: 20 },
  { code: "WP.3", domain: KKSIndicatorDomain.WORK_PERFORMANCE, title: "การพัฒนาตนเองและวิชาชีพ", displayOrder: 30 },
  { code: "WP.4", domain: KKSIndicatorDomain.WORK_PERFORMANCE, title: "การทำงานเป็นทีมและมีส่วนร่วมในสถานศึกษา", displayOrder: 40 },
  { code: "WP.5", domain: KKSIndicatorDomain.WORK_PERFORMANCE, title: "งานที่ได้รับมอบหมาย", displayOrder: 50 },
  { code: "ET.1", domain: KKSIndicatorDomain.ETHICS, title: "วินัยและการรักษาวินัย", displayOrder: 110 },
  { code: "ET.2", domain: KKSIndicatorDomain.ETHICS, title: "คุณธรรม จริยธรรม และจรรยาบรรณวิชาชีพ", displayOrder: 120 },
  { code: "ET.3", domain: KKSIndicatorDomain.ETHICS, title: "ความเป็นแบบอย่างที่ดี", displayOrder: 130 },
];

```

```

for (const it of indicatorSeed) {
    await prisma.kKSIndicator.create({
        data: {
            code: it.code,
            domain: it.domain,
            title: it.title,
            description: null,
            isActive: true,
            displayOrder: it.displayOrder,
        },
    });
}

const allIndicators = await prisma.kKSIndicator.findMany({ select: { id: true, code: true } });
const indId = new Map(allIndicators.map((x) => [x.code, x.id]));

// rules (minimum set)
const ruleSeed = [
    // WP.1 mandatory
    { code: "WP.1", t: EvidenceType.LESSON_PLAN, min: 1, mand: true, note: "อย่างน้อย 1 แผนการสอน" },
    { code: "WP.1", t: EvidenceType.TEACHING_MEDIA, min: 1, mand: true, note: "อย่างน้อย 1 สื่อ/ใบงาน" },
    { code: "WP.1", t: EvidenceType.TEACHING_REFLECTION, min: 1, mand: true, note: "อย่างน้อย 1 บันทึกหลังสอน" },

    // WP.2
    { code: "WP.2", t: EvidenceType.TEACHING_REFLECTION, min: 1, mand: true, note: "สะท้อนการจัดการชั้นเรียน" },
]

```

{ code: "WP.2", t: EvidenceType.ACTIVITY\_PHOTO, min: 1, mand: false, note: "ภาพบรรยากาศชั้นเรียน (แนะนำ)" },

// WP.3

{ code: "WP.3", t: EvidenceType.OFFICIAL\_DOCUMENT, min: 1, mand: true, note: "หลักฐานอบรม/เกียรติบัตร" },

{ code: "WP.3", t: EvidenceType.TEACHING\_REFLECTION, min: 1, mand: false, note: "สรุปสิ่งที่เรียนรู้ (แนะนำ)" },

// WP.4

{ code: "WP.4", t: EvidenceType.ACTIVITY\_PHOTO, min: 1, mand: false, note: "ภาพกิจกรรม/การมีส่วนร่วม (แนะนำ)" },

{ code: "WP.4", t: EvidenceType.OFFICIAL\_DOCUMENT, min: 1, mand: false, note: "คำสั่ง/รายงาน (แนะนำ)" },

// WP.5 mandatory

{ code: "WP.5", t: EvidenceType.OFFICIAL\_DOCUMENT, min: 1, mand: true, note: "คำสั่งมอบหมาย/รายงานผล" },

{ code: "WP.5", t: EvidenceType.ACTIVITY\_PHOTO, min: 1, mand: false, note: "ภาพกิจกรรม (แนะนำ)" },

// ET.1 recommended

{ code: "ET.1", t: EvidenceType.OFFICIAL\_DOCUMENT, min: 1, mand: false, note: "เอกสารรับรอง (แนะนำ)" },

{ code: "ET.1", t: EvidenceType.TEACHING\_REFLECTION, min: 1, mand: false, note: "สะท้อนการรักษาวินัย (แนะนำ)" },

// ET.2 mandatory reflection

{ code: "ET.2", t: EvidenceType.TEACHING\_REFLECTION, min: 1, mand: true, note: "กรณีตัวอย่างเชิงจริยธรรม/ดูแลผู้เรียน" },

```
{ code: "ET.2", t: EvidenceType.ACTIVITY_PHOTO, min: 1, mand: false, note: "กิจกรรมจิตอาสา/  
ดูแลผู้เรียน (แนะนำ)" },
```

```
// ET.3 recommended
```

```
{ code: "ET.3", t: EvidenceType.ACTIVITY_PHOTO, min: 1, mand: false, note: "หลักฐานความเป็นแบบอย่าง (แนะนำ)" },
```

```
{ code: "ET.3", t: EvidenceType.TEACHING_REFLECTION, min: 1, mand: false, note: "สะท้อนตนเอง (แนะนำ)" },
```

```
];
```

```
for (const r of ruleSeed) {
```

```
    const indicatorId = indId.get(r.code);
```

```
    if (!indicatorId) throw new Error(`Missing indicator: ${r.code}`);
```

```
    await prisma.kKSIndicatorEvidenceRequired.create({
```

```
        data: {
```

```
            indicatorId,
```

```
            evidenceType: r.t,
```

```
            minCount: r.min,
```

```
            isMandatory: r.mand,
```

```
            note: r.note,
```

```
        },
```

```
    );
```

```
}
```

```
// 9) AssistantTeacherInfo for assistant teachers in AY2568
```

```
const asstInfo1 = await prisma.assistantTeacherInfo.create({
```

```
    data: {
```

```
        teacherYearId: ty_t1_2568.id,
```

```
        startDate: dt("2025-10-01T00:00:00.000Z"),
```

```
        endDate: dt("2026-09-30T23:59:59.000Z"),
        isCompleted: false,
    },
});

const asstInfo2 = await prisma.assistantTeacherInfo.create({
    data: {
        teacherYearId: ty_t2_2568.id,
        startDate: dt("2025-10-01T00:00:00.000Z"),
        endDate: dt("2026-09-30T23:59:59.000Z"),
        isCompleted: false,
    },
});

// 10) Sample mentoring visits for one assistant teacher
await prisma.mentoringVisit.createMany({
    data: [
        {
            teacherYearId: ty_t1_2568.id,
            visitDate: dt("2025-11-15T00:00:00.000Z"),
            visitType: VisitType.COACHING,
            focusArea: "classroom",
            strengths: "สร้างบรรยายการเรียนรู้ดี",
            challenges: "การประเมินระหว่างเรียนยังไม่สม่ำเสมอ",
            suggestions: "เพิ่ม Exit ticket ทุกคาบ",
            followUpRequired: true,
            observerUserId: mentorA.id,
        },
        {
            teacherYearId: ty_t1_2568.id,
```

```

visitDate: dt("2025-12-10T00:00:00.000Z"),
visitType: VisitType.LESSON_STUDY,
focusArea: "pedagogy",
strengths: "ออกแบบกิจกรรม Active learning ดีมาก",
challenges: "ต้องเพิ่มการสะท้อนหลังสอน",
suggestions: "ทำบันทึกหลังสอนทุกครั้ง",
followUpRequired: false,
observerUserId: mentorA.id,
},
],
});

```

```

// 11) PLC Session + participation
const plc1 = await prisma.pLCSession.create({
  data: {
    academicYearId: ay2568.id,
    plcDate: dt("2025-12-20T00:00:00.000Z"),
    plcLevel: PLCLevel.NETWORK,
    areaId: areaA.id,
    networkGroupId: netA1.id,
    province: "เชียงราย",
    topic: "การวัดและประเมินผลในชั้นเรียน",
    notes: "ใช้เครื่องมือ Exit Ticket / Rubric",
  },
});

```

```

await prisma.pLCParticipation.createMany({
  data: [
    { plcid: plc1.id, teacherYearId: ty_t1_2568.id, role: ParticipationRole.PARTICIPANT, takeaway:
      "นำ Exit ticket ไปใช้จริง" },
  ]
});

```

```
{ plcId: plc1.id, teacherYearId: ty_t2_2568.id, role: ParticipationRole.PARTICIPANT, takeaway:  
"ได้แนว rubric งานเขียน" },  
{ plcId: plc1.id, teacherYearId: ty_t3_2568.id, role: ParticipationRole.PRESENTER, takeaway:  
"แชร์กรณีตัวอย่าง" },  
],  
});
```

// 12) Journals (monthly)

```
await prisma.reflectiveJournal.createMany({  
data: [  
{  
teacherYearId: ty_t1_2568.id,  
month: "2025-11",  
reflectionText: "เริ่มจัดการเรียนรู้แบบกิจกรรมมากขึ้น เด็กมีส่วนร่วมดี",  
successStory: "เด็กกล้าพูดมากขึ้น",  
difficulty: "ประเมินระหว่างเรียนยังไม่ทัน",  
supportRequest: "ขอแบบตัวอย่าง rubric ง่าย ๆ",  
},  
,  
{  
teacherYearId: ty_t1_2568.id,  
month: "2025-12",  
reflectionText: "ลอง Exit ticket และเห็นผลชัด",  
successStory: "รู้ทันที่ว่าเด็กยังไม่เข้าใจตรงไหน",  
difficulty: "เวลาควบเรียนจำกัด",  
supportRequest: "ขอแนวทางจัดเวลา",  
},  
,  
]);
```

// 13) Evidence for assistant teacher t1 (some complete, some missing)

```
const wp1 = indId.get("WP.1")!;
const wp2 = indId.get("WP.2")!;
const wp3 = indId.get("WP.3")!;
const et2 = indId.get("ET.2")!;
const wp5 = indId.get("WP.5")!;

const ev1 = await prisma.teachingEvidence.create({
  data: {
    teacherYearId: ty_t1_2568.id,
    type: EvidenceType.LESSON_PLAN,
    title: "แผนการสอนภาษาไทย ป.3 (หน่วย 2)",
    description: "เน้นอ่านจับใจความ + Active learning",
    fileUrl: "https://demo.local/files/lessonplan_thai_p3_unit2.pdf",
    fileName: "lessonplan_thai_p3_unit2.pdf",
    mimeType: "application/pdf",
    fileSize: 120000,
    indicatorId: wp1,
    round: AssistantEvaluationRound.ROUND_1,
  },
});

const ev2 = await prisma.teachingEvidence.create({
  data: {
    teacherYearId: ty_t1_2568.id,
    type: EvidenceType.TEACHING_REFLECTION,
    title: "บันทึกหลังสอน (พ.ย.)",
    description: "สิ่งที่ได้ผล/สิ่งที่ต้องปรับปรุง",
    fileUrl: "https://demo.local/files/reflection_nov.pdf",
    fileName: "reflection_nov.pdf",
    mimeType: "application/pdf",
  },
});
```

```
    fileSize: 90000,  
    indicatorId: wp1,  
    round: AssistantEvaluationRound.ROUND_1,  
,  
});
```

```
const ev3 = await prisma.teachingEvidence.create({  
  data: {  
    teacherYearId: ty_t1_2568.id,  
    type: EvidenceType.OFFICIAL_DOCUMENT,  
    title: "เกียรติบัตรอบรมการวัดผล",  
    description: "อบรมออนไลน์ 6 ชม.",  
    fileUrl: "https://demo.local/files/certificate_assessment.pdf",  
    fileName: "certificate_assessment.pdf",  
    mimeType: "application/pdf",  
    fileSize: 80000,  
    indicatorId: wp3,  
    round: AssistantEvaluationRound.ROUND_1,  
,  
});
```

```
const ev4 = await prisma.teachingEvidence.create({  
  data: {  
    teacherYearId: ty_t1_2568.id,  
    type: EvidenceType.TEACHING_REFLECTION,  
    title: "กรณีตัวอย่างจรรยาบรรณ (ET.2)",  
    description: "การดูแลผู้เรียนอย่างเป็นธรรม",  
    fileUrl: "https://demo.local/files/ethics_case.pdf",  
    fileName: "ethics_case.pdf",  
    mimeType: "application/pdf",
```

```
    fileSize: 70000,  
    indicatorId: et2,  
    round: AssistantEvaluationRound.ROUND_1,  
,  
});
```

```
const ev5 = await prisma.teachingEvidence.create({  
  data: {  
    teacherYearId: ty_t1_2568.id,  
    type: EvidenceType.OFFICIAL_DOCUMENT,  
    title: "คำสั่งมอบหมายงานกิจกรรมโรงเรียน",  
    description: "รับผิดชอบโครงการอ่านออกเขียนได้",  
    fileUrl: "https://demo.local/files/assignment_order.pdf",  
    fileName: "assignment_order.pdf",  
    mimeType: "application/pdf",  
    fileSize: 65000,  
    indicatorId: wp5,  
    round: AssistantEvaluationRound.ROUND_1,  
,  
});
```

// 14) Assistant Evaluation Round 1 (uses some evidences)

```
const eval1 = await prisma.assistantEvaluation.create({  
  data: {  
    teacherYearId: ty_t1_2568.id,  
    round: AssistantEvaluationRound.ROUND_1,  
    evaluationDate: dt("2026-01-15T00:00:00.000Z"),  
    workPerformancePassed: true,  
    ethicsPassed: true,  
    overallResult: EvaluationResult.PASS,
```

```
remark: "มีพัฒนาการดี เหมาะสมกับวิชาชีพ",
strengths: "ออกแบบกิจกรรมได้ดี นักเรียนมีส่วนร่วม",
improvements: "เพิ่มความสนใจในการประเมินระหว่างเรียน",
createdById: schoolAdminPayaprai.id,
evaluators: {
  create: [
    { evaluatorUserId: schoolAdminPayaprai.id, roleTitle: "ผู้อำนวยการสถานศึกษา", signedAt: dt("2026-01-15T00:00:00.000Z") },
    { evaluatorUserId: mentorA.id, roleTitle: "ศึกษานิเทศก์", signedAt: dt("2026-01-15T00:00:00.000Z") },
    { evaluatorUserId: networkAdminA1.id, roleTitle: "กรรมการ/ผู้แทนเครือข่าย", signedAt: dt("2026-01-15T00:00:00.000Z") },
  ],
},
},
});
```

```
await prisma.assistantEvaluationEvidence.createMany({
  data: [
    { evaluationId: eval1.id, evidenceId: ev1.id },
    { evaluationId: eval1.id, evidenceId: ev2.id },
    { evaluationId: eval1.id, evidenceId: ev3.id },
    { evaluationId: eval1.id, evidenceId: ev4.id },
    { evaluationId: eval1.id, evidenceId: ev5.id },
  ],
});
```

```
// 15) Audit log sample
await prisma.auditLog.createMany({
  data: [
```

```
{ userId: schoolAdminPayaprai.id, action: "CREATE", entity: "AssistantEvaluation", entityId: eval1.id, detail: "Round 1 evaluation created" },  
    { userId: mentorA.id, action: "CREATE", entity: "MentoringVisit", entityId: null, detail: "Coaching visit recorded" },  
],  
});
```

```
console.log(" ✅ Seed complete.");  
console.log("Demo logins (app-level users):");  
console.log("- superadmin@demo.local (SUPER_ADMIN)");  
console.log("- areaadmin.cr3@demo.local (AREA_ADMIN)");  
console.log("- networkadmin.doisung@demo.local (NETWORK_ADMIN)");  
console.log("- director.payaprai@demo.local (SCHOOL_ADMIN)");  
console.log("- mentor.cr3@demo.local (MENTOR)");  
console.log("- assistant1@demo.local (ASSISTANT_TEACHER)");  
}
```

```
main()  
.catch((e) => {  
    console.error(" ❌ Seed failed:", e);  
    process.exit(1);  
})  
.finally(async () => {  
    await prisma.$disconnect();  
});
```

---

คำสั่งใช้งานบน Ubuntu (ให้ Cursor ใส่ใน README ได้เลย)

```
pnpm prisma generate  
pnpm prisma migrate dev  
pnpm prisma db seed
```

pnpm prisma studio

---

ถัดไป (เพื่อให้ Cursor ต่อได้ยาวแบบไม่หลุด) ผนทำให้ต่อได้ทันทีคือ:

- api-spec.md ฉบับ **Multi-tenant + RBAC** (กำหนด filter ตาม role/scope แบบไม่**漏**ข้อมูลข้ามเขต)
- โค้ด endpoint readiness: GET /kks/teachers/:id/readiness?round=ROUND\_1 แบบใช้งานจริง กับ schema ใหม่นี้

เอกสาร api-spec.md ต่อเลยใหม่ครับ?

เอกสาร api-spec.md ต่อเลย

ได้ครับครูใหญ่ — นี่คือ api-spec.md ฉบับ **Multi-tenant + RBAC** ครอบคลุมรับให้ Cursor ทำงานต่อได้ทันที (โครง REST, policies, filters, payloads) โดยออกแบบให้ “**ไม่漏**ข้อมูลข้ามเขต/ข้ามโรงเรียน” และรองรับ multi-year + KKS readiness + PDF export

หมายเหตุ: Auth ใช้ Supabase Auth และ API จะอ้างอิง User ใน DB ของเรา (email/role/scope) ผ่าน authUserId หรือ email mapping

# API Spec: EDU-AREA | Teacher Development & Assistant Evaluation System

Version: 0.1 (MVP)

Base URL: /api

Auth: Supabase Auth (JWT)

DB User Model: public.User (Prisma)

---

## 0) Conventions

### 0.1 Response Format

- Success:

```json

```
{ "data": <any>, "meta": { "requestId": "..." } }
```

- Error:

```
{ "error": { "code": "FORBIDDEN", "message": "..." } }
```

## 0.2 Pagination

List endpoints support:

- page (default 1)
- pageSize (default 20, max 100)
- sort e.g. createdAt:desc

Response meta:

```
{ "meta": { "page": 1, "pageSize": 20, "total": 123 } }
```

## 0.3 Tenant Scope (Important)

All data queries MUST be scoped by RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN: areald required/derived from user
- NETWORK\_ADMIN: areald + networkGroupId
- SCHOOL\_ADMIN: areald + schoolId
- MENTOR: typically areald (and optionally networkGroupId if assigned)
- TEACHER/ASSISTANT\_TEACHER: only own teacherYear records

## 0.4 RBAC helper (Policy)

For each request:

1. Verify Supabase JWT
2. Resolve User in DB:
  - o by authUserId or email
3. Determine scope from user:
  - o areald, networkGroupId, schoolId
4. Apply scope filter to all queries/mutations

---

## 1) Auth & Session

### GET /auth/me

Returns current logged-in user profile + scope.

Response:

```
{
```

```
  "data": {
```

```

"id": "userId",
"email": "mentor.cr3@demo.local",
"displayName": "គីមានិទេក (CR3)",
"role": "MENTOR",
"scope": {
    "areaid": "uuid",
    "networkGroupId": null,
    "schoolId": null
}
}
}

```

#### **POST /auth-sync-user**

Purpose: after Supabase sign-up/sign-in, sync/create DB User profile.

Body:

```
{
    "email": "newuser@x.com",
    "displayName": "ចុះអ្នកខ្សោយ",
    "role": "TEACHER",
    "areaid": "uuid",
    "networkGroupId": "uuid|null",
    "schoolId": "uuid|null",
    "authUserId": "supabase-uuid"
}
```

RBAC:

- SUPER\_ADMIN / AREA\_ADMIN / SCHOOL\_ADMIN can create users in scope
- TEACHER cannot create new users

## **2) Academic Year**

#### **GET /academic-years**

Query: status, isDefault

RBAC:

- All roles can read (scoped not required)

#### **POST /academic-years**

RBAC: SUPER\_ADMIN only

Body:

```
{ "name":"2569", "startDate":"2026-05-01", "endDate":"2027-04-30", "status":"DRAFT", "isDefault":  
false }
```

#### **PATCH /academic-years/:id**

RBAC: SUPER\_ADMIN only

Use to:

- set default year (ensure others set false)
- close year (status=CLOSED)

---

### **3) Organization (Area / Network / School)**

#### **GET /areas**

RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN+: only own area

#### **POST /areas**

RBAC: SUPER\_ADMIN only

#### **GET /areas/:areaid/networks**

RBAC:

- SUPER\_ADMIN: any
- AREA\_ADMIN+: only same area

#### **POST /networks**

RBAC: SUPER\_ADMIN or AREA\_ADMIN (within area)

Body:

```
{ "areaid":"uuid", "code":"NET-X", "name":"กลุ่มเครือข่าย...", "isActive": true }
```

#### **GET /schools**

Filters:

- areald, networkGroupId, province, q (name contains)

RBAC:

- SUPER\_ADMIN: any
- AREA\_ADMIN: areald forced
- NETWORK\_ADMIN: networkGroupId forced
- SCHOOL\_ADMIN: only own school

#### POST /schools

RBAC:

- SUPER\_ADMIN / AREA\_ADMIN: create in area
- NETWORK\_ADMIN: create in own network

Body:

```
{  
    "areald": "uuid",  
    "networkGroupId": "uuid|null",  
    "code": "SCH-XXX",  
    "schoolName": "...",  
    "province": "...",  
    "region": "NORTH",  
    "schoolSize": "SMALL",  
    "areaType": "REMOTE",  
    "studentTotal": 120  
}
```

---

#### 4) Users (RBAC Management)

##### GET /users

Filters: role, areald, networkGroupId, schoolId, q

RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN: users in area

- NETWORK\_ADMIN: users in network
- SCHOOL\_ADMIN: users in school
- Others: forbidden

#### **POST /users**

RBAC: same as above

Creates DB user (and optionally later links authUserId)

#### **PATCH /users/:id**

RBAC:

- can update users within own scope
- cannot escalate role above self

## 5) Teachers & Teacher-Year Records

#### **GET /teachers**

Filters:

- areaId, networkGroupId, schoolId
- academicYearId (optional; if provided join TeacherYearRecord)
- isAssistantTeacher (true/false with academicYearId)

RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN: in area
- NETWORK\_ADMIN: in network
- SCHOOL\_ADMIN: in school
- MENTOR: in area (optionally restricted to assigned networks later)
- TEACHER/ASSISTANT\_TEACHER: only self

#### **POST /teachers**

RBAC:

- SCHOOL\_ADMIN / AREA\_ADMIN / NETWORK\_ADMIN (within scope)

Body:

{

  "fullName": "...",

```
"citizenId": "...",
"appointmentDate": "2025-10-01",
"position": "គ្រួសារ",
"major": "...",
"areald": "uuid",
"networkGroupId": "uuid|null",
"schoolId": "uuid",
"userId": "uuid|null"
}
```

#### **POST /teachers/:teacherId/year-records**

Create TeacherYearRecord for a given academic year.

RBAC: admin in scope only

Body:

```
{
  "academicYearId": "uuid",
  "cohort": 1,
  "isKruRakThin": true,
  "isAssistantTeacher": true
}
```

#### **GET /teacher-years/:id**

Returns TeacherYearRecord full detail (visits, journals, evidences summary)

RBAC:

- Admin/mentor in scope OR teacher self

---

## 6) Mentoring Visits

#### **GET /mentoring-visits**

Filters:

- teacherYearId
- from, to

- visitType  
RBAC: scoped

#### **POST /mentoring-visits**

RBAC:

- MENTOR / SCHOOL\_ADMIN / NETWORK\_ADMIN / AREA\_ADMIN in scope

Body:

```
{  
  "teacherYearId": "uuid",  
  "visitDate": "2026-01-10",  
  "visitType": "COACHING",  
  "focusArea": "classroom",  
  "strengths": "...",  
  "challenges": "...",  
  "suggestions": "...",  
  "followUpRequired": true  
}
```

---

## 7) Reflective Journals

#### **GET /journals**

Filters: teacherYearId, month

RBAC:

- Teacher self OR mentor/admin in scope

#### **POST /journals**

RBAC:

- Teacher self

Body:

```
{  
  "teacherYearId": "uuid",  
  "month": "2026-01",  
  "reflectionText": "...",  
  "rating": 5  
}
```

```
"successStory": "...",
"difficulty": "...",
"supportRequest": "..."
}
```

---

## 8) PLC

### **GET /plc/sessions**

Filters: academicYearId, areald, networkGroupId, from, to, q

RBAC: scoped

### **POST /plc/sessions**

RBAC: MENTOR/ADMIN in scope

Body:

```
{
  "academicYearId": "uuid",
  "plcDate": "2026-01-20",
  "plcLevel": "NETWORK",
  "areald": "uuid",
  "networkGroupId": "uuid",
  "topic": "...",
  "notes": "..."
}
```

### **POST /plc/participations**

RBAC: MENTOR/ADMIN in scope OR teacher self (optional)

Body:

```
{
  "plcid": "uuid",
  "teacherYearId": "uuid",
  "role": "PARTICIPANT",
  "takeaway": "..."
}
```

---

## 9) KKS Indicators & Rules (Global Config)

### GET /kks/indicators

Returns active indicators + evidence rules.

RBAC: all logged-in

Response:

```
{  
  "data": [  
    {  
      "id": "uuid",  
      "code": "WP.1",  
      "domain": "WORK_PERFORMANCE",  
      "title": "การจัดการเรียนรู้",  
      "displayOrder": 10,  
      "evidenceRules": [  
        { "evidenceType": "LESSON_PLAN", "minCount": 1, "isMandatory": true, "note": "..." }  
      ]  
    }  
  ]  
}
```

### POST /kks/indicators

RBAC: SUPER\_ADMIN only

Used for config changes

### POST /kks/indicators/:id/rules

RBAC: SUPER\_ADMIN only

---

## 10) Evidence / Portfolio

### GET /evidences

Filters:

- teacherYearId (required unless SUPER\_ADMIN with broad query)

- indicatorCode or indicatorId
  - type
  - round
- RBAC: teacher self or scope admin

#### **POST /evidences**

RBAC: teacher self OR admin in scope (for assisting upload)

Body:

```
{  
  "teacherYearId": "uuid",  
  "indicatorId": "uuid",  
  "type": "LESSON_PLAN",  
  "title": "แผนการสอน...",  
  "description": "...",  
  "fileUrl": "https://...",  
  "fileName": "...",  
  "mimeType": "application/pdf",  
  "fileSize": 12345,  
  "round": "ROUND_1"  
}
```

#### **DELETE /evidences/:id**

RBAC:

- owner teacher OR scope admin
- record audit log

---

#### **11) Readiness Engine (Assistant Teacher)**

#### **GET /kks/teachers/:teacherYearId/readiness?round=ROUND\_1**

Purpose:

- For a teacher-year record, compute missing mandatory evidence by indicator.

RBAC:

- Teacher self OR mentor/admin in scope

Algorithm:

1. fetch active indicators + rules (mandatory)
2. count evidences where teacherYearId AND (round matches if provided) grouped by indicatorId + type
3. for each indicator mandatory rule: if have < minCount => missing
4. overallReady = missing.length == 0

Response:

```
{  
  "data":{  
    "teacherYearId":"uuid",  
    "round":"ROUND_1",  
    "overallReady": false,  
    "missing": [  
      {  
        "indicatorCode": "WP.1",  
        "indicatorTitle": "การจัดการเรียนรู้",  
        "missingRules": [  
          { "evidenceType": "TEACHING_MEDIA", "required": 1, "have": 0, "isMandatory": true, "note": "..." }  
        ]  
      }  
    ],  
    "summaryCounts": {  
      "mandatoryIndicatorsReady": 6,  
      "mandatoryIndicatorsTotal": 8  
    }  
  }  
}
```

---

## 12) Assistant Teacher Info & Evaluations

**POST /assistant-teachers/:teacherYearId/start**

RBAC: SCHOOL\_ADMIN / AREA\_ADMIN / NETWORK\_ADMIN in scope

Body:

```
{ "startDate":"2025-10-01", "endDate":"2026-09-30" }
```

**GET /assistant-evaluations**

Filters: teacherYearId, round

RBAC: teacher self or admin in scope

**POST /assistant-evaluations**

RBAC: SCHOOL\_ADMIN/MENTOR/NETWORK\_ADMIN/AREA\_ADMIN in scope

Body:

```
{
  "teacherYearId": "uuid",
  "round": "ROUND_1",
  "evaluationDate": "2026-01-15",
  "workPerformancePassed": true,
  "ethicsPassed": true,
  "overallResult": "PASS",
  "remark": "...",
  "strengths": "...",
  "improvements": "...",
  "evaluatorUserIds": ["uuid1", "uuid2", "uuid3"],
  "usedEvidenceIds": ["e1", "e2", "e3"]
}
```

**GET /assistant-evaluations/:id**

Returns evaluation + evaluators + used evidences.

**POST /assistant-evaluations/:id/sign**

RBAC: evaluator only

Body:

```
{ "signedAt": "2026-01-15T10:00:00Z" }
```

---

**13) Dashboard (Status for Director/Area)**

GET /dashboards/assistant-teachers/overview?academicYearId=...&round=ROUND\_1

RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN: own area
- NETWORK\_ADMIN: own network
- SCHOOL\_ADMIN: own school
- MENTOR: own area

Returns:

- counts (ready/not ready/urgent)
- lists (top missing indicators)
- (optional) heatmap matrix data

Response (minimal):

```
{  
  "data": {  
    "counts": { "total": 12, "ready": 7, "notReady": 5, "urgent": 2 },  
    "topMissing": [  
      { "indicatorCode": "WP.1", "missingCount": 3 },  
      { "indicatorCode": "ET.2", "missingCount": 2 }  
    ],  
    "notReadyList": [  
      {  
        "teacherYearId": "uuid",  
        "teacherName": "...",  
        "schoolName": "...",  
        "missingSummary": "WP.1: TEACHING_MEDIA; ET.2: TEACHING_REFLECTION"  
      }  
    ]  
  }  
}
```

---

## 14) Report Export (PDF/Excel)

**POST /reports/export**

RBAC: admin/mentor in scope

Body:

```
{  
  "academicYearId": "uuid",  
  "scopeType": "SCHOOL",  
  "scopeId": "uuid",  
  "reportType": "ASSISTANT_STATUS",  
  "round": "ROUND_1"  
}
```

Behavior:

- generate file (async ok)
- store ReportExport record
- return fileUrl when ready

**GET /reports**

RBAC: scoped

Returns export history.

---

## 15) Audit Log

**GET /audit-logs**

RBAC:

- SUPER\_ADMIN: all
- AREA\_ADMIN: within area
- SCHOOL\_ADMIN: within school

Filters: entity, action, from, to, userId

---

## RBAC Rules (Implementation Notes for Cursor)

### ### Scope filter templates

- For resources linked to TeacherYearRecord:
  - join teacherYear.teacher.schoolId/areaid/networkGroupId and enforce scope
- For resources linked to School:
  - enforce areaid/networkGroupId/schoolId as per user

### ### Prevent data leakage

- If role != SUPER\_ADMIN, ignore query params that attempt broader scope
  - e.g., AREA\_ADMIN cannot request areaid != own areaid

### ### Audit Logging

Write AuditLog on:

- login/sync-user
- create/update/delete evidence
- create evaluation/sign
- export reports

---

## ## Next Recommended Tasks

- 1) Implement `/auth/me` and DB user resolution
- 2) Implement scope middleware (RBAC guard)
- 3) Implement `GET /kks/indicators`
- 4) Implement `GET /kks/teachers/:teacherYearId/readiness`
- 5) Build dashboard overview endpoint

---

ถัดไปผมเสนอ “ไฟล์เพิ่มอีก 1 ชิ้น” ที่ช่วย Cursor ทำงานเร็วมาก:

**rbac.ts (policy helpers)** + ตัวอย่าง Prisma where-filter ตาม role/scope (กันหลุดแบบ 0%)  
เอกสารอื่นๆ ใหมครับ?

ເອກຕົວເລຍ

ໄດ້ຄັບຄຽງໃໝ່  ນີ້ຄື່ອື່ນີ້ແລ້ວ src/lib/rbac.ts (ຫຼືຈະວາງທີ່ src/server/rbac.ts ກີ່ໄດ້) ສໍາຮັບ RBAC + Multi-tenant scope filtering ແບບ “ກັນຫຼຸດ 0%” ໂດຍອອກແບບໃຫ້ເຊີ້ນ Next.js Route Handlers ໄດ້ທັນທີ ລັກຄິດ: ໂມ່ເຊື່ອ query params ຕ້າຜູ້ໃໝ່ມີສີທີ່ — ບັນດັບ scope ຈາກ DB user ເທົ່ານັ້ນ

```
// src/lib/rbac.ts
```

```
import { Prisma } from "@prisma/client";
import type { User, UserRole } from "@prisma/client";
```

```
export type Scope = {
    areald?: string | null;
    networkGroupId?: string | null;
    schoolId?: string | null;
};
```

```
export type Actor = Pick<User, "id" | "email" | "role" | "areald" | "networkGroupId" | "schoolId" |
    "isActive">;
```

```
export class RBACError extends Error {
    code: "UNAUTHENTICATED" | "FORBIDDEN" | "BAD_REQUEST";
    constructor(code: RBACError["code"], message: string) {
        super(message);
        this.code = code;
    }
}
```

```
/**
 * Validate actor is active and has minimal required scope for certain roles
 */
export function assertActorActive(actor: Actor) {
    if (!actor?.id) throw new RBACError("UNAUTHENTICATED", "No actor");
```

```

if (!actor.isActive) throw new RBACError("FORBIDDEN", "User is inactive");
}

/***
 * Some roles must have an areald/schoold to operate meaningfully.
 * SUPER_ADMIN can be global with no scope.
 */
export function assertActorHasScope(actor: Actor) {
  if (actor.role === "SUPER_ADMIN") return;

  // AREA_ADMIN / MENTOR should at least have areald
  if ((actor.role === "AREA_ADMIN" || actor.role === "MENTOR") && !actor.areald) {
    throw new RBACError("FORBIDDEN", "Missing area scope");
  }

  // NETWORK_ADMIN should have areald + networkGroupId
  if (actor.role === "NETWORK_ADMIN" && (!actor.areald || !actor.networkGroupId)) {
    throw new RBACError("FORBIDDEN", "Missing network scope");
  }

  // SCHOOL_ADMIN / TEACHER / ASSISTANT_TEACHER should have areald + schoold
  if (
    (actor.role === "SCHOOL_ADMIN" || actor.role === "TEACHER" || actor.role ===
    "ASSISTANT_TEACHER") &&
    (!actor.areald || !actor.schoold)
  ) {
    throw new RBACError("FORBIDDEN", "Missing school scope");
  }
}

```

```

/**
 * Generic scope object from actor
 */
export function actorScope(actor: Actor): Scope {
    return {
        areald: actor.areald ?? null,
        networkGroupId: actor.networkGroupId ?? null,
        schoolId: actor.schoolId ?? null,
    };
}

/***
 * Access checks for a given target scope.
 * Use when request includes a specific area/network/school target.
 */
export function canAccessScope(actor: Actor, target: Scope): boolean {
    if (actor.role === "SUPER_ADMIN") return true;

    // If actor has areald, must match target.areald (when target has one)
    if (target.areald && actor.areald && target.areald !== actor.areald) return false;

    // NETWORK_ADMIN locked to networkGroupId
    if (actor.role === "NETWORK_ADMIN") {
        if (target.networkGroupId && actor.networkGroupId && target.networkGroupId !==
            actor.networkGroupId) return false;
        // If target doesn't specify networkGroupId but is broader than actor's network, deny by
        default
        if (!target.networkGroupId) return false;
        return true;
    }
}

```

```

// SCHOOL_ADMIN / TEACHER / ASSISTANT_TEACHER locked to schoolId
if (actor.role === "SCHOOL_ADMIN" || actor.role === "TEACHER" || actor.role ===
"ASSISTANT_TEACHER") {
    if (target.schoolId && actor.schoolId && target.schoolId !== actor.schoolId) return false;
    if (!target.schoolId) return false;
    return true;
}

// AREA_ADMIN / MENTOR: can access within area
if (actor.role === "AREA_ADMIN" || actor.role === "MENTOR") {
    // If target asks for another area => already filtered above
    return true;
}

return false;
}

/***
 * Force scope filters for School queries.
 * Ignores client query params if user isn't allowed broader scope.
 */
export function schoolWhere(actor: Actor, requested?: Scope): Prisma.SchoolWhereInput {
    // SUPER_ADMIN can use requested filters as-is
    if (actor.role === "SUPER_ADMIN") {
        return {
            arealId: requested?.arealId ?? undefined,
            networkGroupId: requested?.networkGroupId ?? undefined,
        };
    }
}

```

```

// Everyone else: force from actor scope
const base: Prisma.SchoolWhereInput = {
    areald: actor.areald ?? undefined,
};

if (actor.role === "NETWORK_ADMIN") {
    base.networkGroupId = actor.networkGroupId ?? undefined;
    return base;
}

if (actor.role === "SCHOOL_ADMIN" || actor.role === "TEACHER" || actor.role ===
    "ASSISTANT_TEACHER") {
    base.id = actor.schoolId ?? undefined;
    return base;
}

// AREA_ADMIN / MENTOR: can optionally filter by networkGroupId if requested (still within
area)
if (requested?.networkGroupId) base.networkGroupId = requested.networkGroupId;
return base;
}

/**
 * Teacher queries are constrained by Teacher.areald / networkGroupId / schoolId
 */
export function teacherWhere(actor: Actor, requested?: Scope): Prisma.TeacherWhereInput {
    if (actor.role === "SUPER_ADMIN") {
        return {
            areald: requested?.areald ?? undefined,

```

```

    networkGroupId: requested?.networkGroupId ?? undefined,
    schoolId: requested?.schoolId ?? undefined,
};

}

const base: Prisma.TeacherWhereInput = {
    areaId: actor.areaId ?? undefined,
};

if (actor.role === "NETWORK_ADMIN") {
    base.networkGroupId = actor.networkGroupId ?? undefined;
    // still allow optional schoolId filter if within same network (client provided)
    if (requested?.schoolId) base.schoolId = requested.schoolId;
    return base;
}

if (actor.role === "SCHOOL_ADMIN") {
    base.schoolId = actor.schoolId ?? undefined;
    return base;
}

if (actor.role === "MENTOR" || actor.role === "AREA_ADMIN") {
    // allow requested narrowing but never broadening beyond area
    if (requested?.networkGroupId) base.networkGroupId = requested.networkGroupId;
    if (requested?.schoolId) base.schoolId = requested.schoolId;
    return base;
}

// TEACHER/ASSISTANT_TEACHER: only self (via userId linkage)
base.userId = actor.id;

```

```

return base;
}

/** 
 * TeacherYearRecord queries constrained by teacher scope.
 * Works by nested relation: TeacherYearRecord.teacher -> (area/network/school/user)
 */
export function teacherYearWhere(actor: Actor, requested?: Scope):
Prisma.TeacherYearRecordWhereInput {
if (actor.role === "SUPER_ADMIN") {
  return {
    teacher: {
      arealId: requested?.arealId ?? undefined,
      networkGroupId: requested?.networkGroupId ?? undefined,
      schoolId: requested?.schoolId ?? undefined,
    },
  };
}

const baseTeacher: Prisma.TeacherWhereInput = {
  arealId: actor.arealId ?? undefined,
};

if (actor.role === "NETWORK_ADMIN") {
  baseTeacher.networkGroupId = actor.networkGroupId ?? undefined;
  if (requested?.schoolId) baseTeacher.schoolId = requested.schoolId;
} else if (actor.role === "SCHOOL_ADMIN") {
  baseTeacher.schoolId = actor.schoolId ?? undefined;
} else if (actor.role === "MENTOR" || actor.role === "AREA_ADMIN") {
  if (requested?.networkGroupId) baseTeacher.networkGroupId = requested.networkGroupId;
}

```

```

if (requested?.schoolId) baseTeacher.schoolId = requested.schoolId;
} else {
    // teacher self
    baseTeacher.userId = actor.id;
}

return { teacher: baseTeacher };
}

/**
 * Assert the actor can access a specific TeacherYearRecord by scope.
 * Use when endpoint param contains teacherYearId.
 */
export async function assertCanAccessTeacherYear(prisma: any, actor: Actor, teacherYearId: string) {
    assertActorActive(actor);
    assertActorHasScope(actor);

    const record = await prisma.teacherYearRecord.findUnique({
        where: { id: teacherYearId },
        select: {
            id: true,
            teacher: {
                select: { arealId: true, networkGroupId: true, schoolId: true, userId: true },
            },
        },
    });

    if (!record) throw new RBACError("BAD_REQUEST", "TeacherYearRecord not found");
}

```

```

// Self access

if ((actor.role === "TEACHER" || actor.role === "ASSISTANT_TEACHER") && record.teacher.userId
!== actor.id) {
    throw new RBACError("FORBIDDEN", "Not your record");
}

// Admin/mentor access via scope

const ok = canAccessScope(actor, {
    areald: record.teacher.areald,
    networkGroupId: record.teacher.networkGroupId,
    schoolId: record.teacher.schoolId,
});

if (!ok) throw new RBACError("FORBIDDEN", "Out of scope");

}

/***
 * Utility: sanitize requested scope from query params
 * Only keep strings; empty => undefined
 */
export function parseScopeFromQuery(qs: URLSearchParams): Scope {
    const val = (k: string) => {
        const v = qs.get(k);
        return v && v.trim().length > 0 ? v.trim() : undefined;
    };
    return {
        areald: val("areald"),
        networkGroupId: val("networkGroupId"),
        schoolId: val("schoolId"),
    };
}

```

```

}

/***
 * Helper: convert RBACError to API response
 */
export function rbacErrorToResponse(e: any) {
  if (e instanceof RBACError) {
    const status =
      e.code === "UNAUTHENTICATED" ? 401 :
      e.code === "FORBIDDEN" ? 403 : 400;

    return new Response(JSON.stringify({ error: { code: e.code, message: e.message } }), {
      status,
      headers: { "Content-Type": "application/json; charset=utf-8" },
    });
  }

  return new Response(JSON.stringify({ error: { code: "INTERNAL", message: "Unexpected error" } }), {
    status: 500,
    headers: { "Content-Type": "application/json; charset=utf-8" },
  });
}

```

### ตัวอย่างการใช้ใน Next.js Route Handler (ลื้น ๆ แต่ครบ)

เอกสารใน app/api/kks/teachers/[teacherYearId]/readiness/route.ts

```

import { prisma } from "@/lib/prisma";
import { getActorFromRequest } from "@/lib/auth"; // ให้ Cursor เจียน: resolve Supabase JWT ->
DB user
import { assertCanAccessTeacherYear, rbacErrorToResponse } from "@/lib/rbac";

```

```

export async function GET(req: Request, ctx: { params: { teacherYearId: string } }) {
  try {
    const actor = await getActorFromRequest(req);
    await assertCanAccessTeacherYear(prisma, actor, ctx.params.teacherYearId);

    // ... compute readiness here ...
    return Response.json({ data: { ok: true } });
  } catch (e) {
    return rbacErrorToResponse(e);
  }
}

```

### จุดที่ Cursor ต้องทำต่อ (แต่โครงขั้ดแล้ว)

1. src/lib/auth.ts → getActorFromRequest(req)
  - verify Supabase JWT
  - map email/authUserId -> Prisma User
2. src/lib/prisma.ts → prisma singleton
3. Implement endpoints ตาม api-spec โดยเรียก  
teacherWhere/teacherYearWhere/schoolWhere

ถ้าครูใหญ่เอาจริงแบบ “ให้ Cursor วิ่งเร็วแบบติดเทอร์โบ” ผมทำต่อให้เลยเป็น:

- auth.ts (Supabase JWT verify + DB user mapping)
- prisma.ts singleton
- ตัวอย่าง endpoint จริง 2 ตัว:
  - GET /kks/indicators
  - GET /kks/teachers/:teacherYearId/readiness (ทำงานครับ + response ตามสเปค)

ให้ผม “ลงโค้ด auth.ts + prisma.ts + 2 endpoints” ต่อเลยไหมครับ?

“ลงโค้ด auth.ts + prisma.ts + 2 endpoints” ต่อเลย

Thought for 9s

ได้เลยครูใหญ่ ✅ ผลงานโค้ดให้ครบชุดตามที่ขอ:

- src/lib/prisma.ts (Prisma singleton สำหรับ Next.js)
- src/lib/auth.ts (อ่าน Bearer token → verify กับ Supabase → map เป็น DB User → return actor)
- 2 endpoints:
  - app/api/kks/indicators/route.ts
  - app/api/kks/teachers/[teacherYearId]/readiness/route.ts

หมายเหตุสั้น ๆ แบบไม่กวนใจ: Next.js 15 โดยปกติใช้ Node รุ่นใหม่กว่านี้ (Node 18+) และโค้ดชุดนี้เขียนแบบ “server standard” ใช้เด็กับ Ubuntu + Supabase + Prisma ตามที่ออกแบบไว้ครับ

---

### 1) src/lib/prisma.ts

```
// src/lib/prisma.ts

import { PrismaClient } from "@prisma/client";

declare global {
    // eslint-disable-next-line no-var
    var __prisma: PrismaClient | undefined;
}

/**
 * Prisma singleton for Next.js (prevents exhausting DB connections in dev hot reload)
 */

export const prisma =
    global.__prisma ??
    new PrismaClient({
        log: process.env.NODE_ENV === "development" ? ["warn", "error"] : ["error"],
    });

if (process.env.NODE_ENV !== "production") {
    global.__prisma = prisma;
```

```
}
```

---

## 2) src/lib/auth.ts

ต้องตั้ง env:

- SUPABASE\_URL
- SUPABASE\_ANON\_KEY (หรือใช้ SERVICE\_ROLE\_KEY ก็ได้ ถ้าต้องการความเสถียรสูงในการตรวจ token)
- (DB) Prisma เชื่อมผ่าน DATABASE\_URL

```
// src/lib/auth.ts
```

```
import { createClient } from "@supabase/supabase-js";
import type { Actor } from "@/lib/rbac";
import { RBACError, assertActorActive, assertActorHasScope } from "@/lib/rbac";
import { prisma } from "@/lib/prisma";
```

```
// Use anon key for getUser(token). If you want stronger admin-level verification,
```

```
// you can use SUPABASE_SERVICE_ROLE_KEY instead (server-only).
```

```
const SUPABASE_URL = process.env.SUPABASE_URL;
```

```
const SUPABASE_KEY =
```

```
process.env.SUPABASE_SERVICE_ROLE_KEY ||
```

```
process.env.SUPABASE_ANON_KEY ||
```

```
"";
```

```
if (!SUPABASE_URL) throw new Error("Missing env SUPABASE_URL");
```

```
if (!SUPABASE_KEY) throw new Error("Missing env SUPABASE_ANON_KEY or
SUPABASE_SERVICE_ROLE_KEY");
```

```
const supabase = createClient(SUPABASE_URL, SUPABASE_KEY, {
  auth: { persistSession: false, autoRefreshToken: false },
});
```

```
function getBearerToken(req: Request): string | null {
  const h = req.headers.get("authorization") || req.headers.get("Authorization");
  if (!h) return null;
  const m = h.match(/^Bearer\s+(.+)$/i);
  return m?.[1] ?? null;
}

/**
 * Resolve actor (DB User) from Supabase JWT in Authorization: Bearer <token>
 * Policy:
 * - Must be valid Supabase user
 * - Must exist in DB public.User (seed or created via sync-user)
 */
export async function getActorFromRequest(req: Request): Promise<Actor> {
  const token = getBearerToken(req);
  if (!token) throw new RBACError("UNAUTHENTICATED", "Missing bearer token");

  // Verify token by calling Supabase Auth
  const { data, error } = await supabase.auth.getUser(token);
  if (error || !data?.user) {
    throw new RBACError("UNAUTHENTICATED", "Invalid or expired token");
  }

  const authUserId = data.user.id;
  const email = data.user.email;

  if (!email) throw new RBACError("UNAUTHENTICATED", "Supabase user missing email");

  // Find DB user by authUserId first, fallback to email
  const dbUser =
```

```

(await prisma.user.findUnique({ where: { authUserId }, select: actorSelect })) ||
(await prisma.user.findUnique({ where: { email }, select: actorSelect }));

if (!dbUser) {
  // Do NOT auto-create here by default (safer).
  // Use /auth-sync-user or admin provisioning.
  throw new RBACError("FORBIDDEN", "User not provisioned in system");
}

// If found by email but authUserId not linked yet, link it (safe improvement)
if (!dbUser.authUserId) {
  await prisma.user.update({
    where: { id: dbUser.id },
    data: { authUserId },
  });
  // return updated actor
  const updated = await prisma.user.findUnique({ where: { id: dbUser.id }, select: actorSelect });
  if (!updated) throw new RBACError("FORBIDDEN", "User provisioning issue");
  assertActorActive(updated);
  assertActorHasScope(updated);
  return updated;
}

assertActorActive(dbUser);
assertActorHasScope(dbUser);
return dbUser;
}

const actorSelect = {
  id: true,

```

```
authUserId: true,  
email: true,  
displayName: true,  
role: true,  
isActive: true,  
areald: true,  
networkGroupId: true,  
schoolId: true,  
} as const;
```

---

### 3) Endpoint: GET /api/kks/indicators

```
Path: app/api/kks/indicators/route.ts  
  
// app/api/kks/indicators/route.ts  
  
import { prisma } from "@/lib/prisma";  
import { getActorFromRequest } from "@/lib/auth";  
import { rbacErrorToResponse } from "@/lib/rbac";  
  
export async function GET(req: Request) {  
  try {  
    await getActorFromRequest(req); // any logged-in role can read  
  
    const indicators = await prisma.kKSIndicator.findMany({  
      where: { isActive: true },  
      orderBy: [{ domain: "asc" }, { displayOrder: "asc" }, { code: "asc" }],  
      include: {  
        evidenceRules: {  
          orderBy: [{ isMandatory: "desc" }, { evidenceType: "asc" }],  
          select: {  
            evidenceType: true,  
            minCount: true,  
          }  
        }  
      }  
    });  
  } catch (err) {  
    return rbacErrorToResponse(err);  
  }  
  return res.json(indicators);  
}  
};
```

```

    isMandatory: true,
    note: true,
  },
},
},
});
};

return Response.json({ data: indicators, meta: {} }, { status: 200 });
} catch (e) {
  return rbacErrorToResponse(e);
}
}

```

---

#### 4) Endpoint: GET /api/kks/teachers/:teacherYearId/readiness?round=ROUND\_1

Path: app/api/kks/teachers/[teacherYearId]/readiness/route.ts

```

// app/api/kks/teachers/[teacherYearId]/readiness/route.ts

import { prisma } from "@/lib/prisma";
import { getActorFromRequest } from "@/lib/auth";
import { assertCanAccessTeacherYear, rbacErrorToResponse } from "@/lib/rbac";
import { AssistantEvaluationRound } from "@prisma/client";

function parseRound(qs: URLSearchParams): AssistantEvaluationRound | null {
  const r = qs.get("round");
  if (!r) return null;
  if (r === "ROUND_1") return "ROUND_1";
  if (r === "ROUND_2") return "ROUND_2";
  return null;
}

export async function GET(req: Request, ctx: { params: { teacherYearId: string } }) {

```

```
try {

  const actor = await getActorFromRequest(req);
  const teacherYearId = ctx.params.teacherYearId;

  await assertCanAccessTeacherYear(prisma, actor, teacherYearId);

  const url = new URL(req.url);
  const round = parseRound(url.searchParams); // optional filter

  // 1) fetch indicators + rules (only mandatory rules are used for readiness)
  const indicators = await prisma.kKSIndicator.findMany({
    where: { isActive: true },
    orderBy: [{ domain: "asc" }, { displayOrder: "asc" }, { code: "asc" }],
    include: {
      evidenceRules: {
        where: { isMandatory: true },
        select: {
          evidenceType: true,
          minCount: true,
          isMandatory: true,
          note: true,
        },
      },
    },
  });

  // 2) fetch evidences grouped by indicatorId + type (and round if specified)
  const evidences = await prisma.teachingEvidence.findMany({
    where: {
      teacherYearId,
```

```

...(round ? { round } : {}),
},
select: {
  indicatorId: true,
  type: true,
},
});

const haveCount = new Map<string, number>(); // key = `${indicatorId}:${type}`
for (const e of evidences) {
  const k = `${e.indicatorId}:${e.type}`;
  haveCount.set(k, (haveCount.get(k) ?? 0) + 1);
}

// 3) compute missing
const missing: Array<{
  indicatorId: string;
  indicatorCode: string;
  indicatorTitle: string;
  missingRules: Array<{
    evidenceType: string;
    required: number;
    have: number;
    isMandatory: boolean;
    note: string | null;
  }>;
}> = [];

let mandatoryIndicatorsTotal = 0;
let mandatoryIndicatorsReady = 0;

```

```
for (const ind of indicators) {
    if (!ind.evidenceRules || ind.evidenceRules.length === 0) continue;

    mandatoryIndicatorsTotal++;

    const missingRules: any[] = [];
    for (const rule of ind.evidenceRules) {
        const key = `${ind.id}:${rule.evidenceType}`;
        const have = haveCount.get(key) ?? 0;
        if (have < rule.minCount) {
            missingRules.push({
                evidenceType: rule.evidenceType,
                required: rule.minCount,
                have,
                isMandatory: true,
                note: rule.note ?? null,
            });
        }
    }

    if (missingRules.length === 0) {
        mandatoryIndicatorsReady++;
    } else {
        missing.push({
            indicatorId: ind.id,
            indicatorCode: ind.code,
            indicatorTitle: ind.title,
            missingRules,
        });
    }
}
```

```

        }
    }

const overallReady = missing.length === 0;

return Response.json(
{
    data: {
        teacherYearId,
        round: round ?? null,
        overallReady,
        missing,
        summaryCounts: {
            mandatoryIndicatorsReady,
            mandatoryIndicatorsTotal,
        },
    },
    meta: {},
},
{ status: 200 });
};

} catch (e) {
    return rbacErrorToResponse(e);
}
}

```

เช็กลิสต์ env ที่ต้องมี (ใส่ใน .env.local)

```

DATABASE_URL="postgresql://..."
SUPABASE_URL="https://xxxxx.supabase.co"
SUPABASE_ANON_KEY="xxxxx"

```

```
# (optional, server-only)  
SUPABASE_SERVICE_ROLE_KEY="xxxxx"
```

---

### คำสั่งทดสอบเร็ว (Curl)

```
# token = Supabase access_token  
curl -H "Authorization: Bearer $TOKEN" http://localhost:3000/api/kks/indicators  
  
curl -H "Authorization: Bearer $TOKEN" \  
"http://localhost:3000/api/kks/teachers/<teacherYearId>/readiness?round=ROUND_1"
```

---

ถัดไปผมทำให้ “จบเป็นชุดใช้งานจริง” ได้เลยอีก 2 ชิ้น (แนะนำมาก):

1. GET /api/dashboards/assistant-teachers/overview (สรุป ready/not ready/top missing ตาม scope)
2. app/api/auth/me + app/api/auth/sync-user (provision ผู้ใช้เข้าระบบแบบราชการ)

ให้ผมลง 2 endpoint นี้ต่อเลยไหมครับ?

PDF Template (TH Sarabun / A4 / ตรารูฐ)

จัดให้แบบ ราชการเป็น ๆ ใช้งานได้จริง ครบครูในที่ TH

ผมให้มาเป็น HTML → PDF (A4 / TH Sarabun / ตรารูฐ) ซึ่งหมายความว่า สุดกับ Next.js + Node (ใช้ Puppeteer) และแก้ไขฟอร์มได้ง่ายในอนาคต

แนวคิด:

- หน้าตา = แบบฟอร์มราชการ 100%
- โครงสร้าง = อิง AssistantEvaluation + TeacherYearRecord + School
- ใช้ได้ทั้ง “พิมพ์” และ “แนบระบบ”

---

### โครงสร้างไฟล์ที่แนะนำ

```
/templates/pdf/  
|—— assistant-evaluation.html  
|—— th-sarabun.css  
└—— garuda-base64.txt (รูปครูท base64)
```

และไฟล์ generate:

```
/src/lib/pdf/  
└── generateAssistantEvaluationPDF.ts
```

---



### ฟอนต์ราชการ: TH Sarabun

**templates/pdf/th-sarabun.css**

```
@font-face {  
    font-family: 'THSarabun';  
    src: url('https://fonts.gstatic.com/s/sarabun/v15/DtVjJx26TKEr37c9YHZJpA.ttf')  
        format('truetype');  
    font-weight: normal;  
}  
  
@font-face {  
    font-family: 'THSarabun';  
    src: url('https://fonts.gstatic.com/s/sarabun/v15/DtVhJx26TKEr37c9YOZJpq4.ttf')  
        format('truetype');  
    font-weight: bold;  
}  
  
body {  
    font-family: 'THSarabun', serif;  
    font-size: 16pt;  
    line-height: 1.2;  
}
```

---

### 2 HTML Template (A4 + ตราครุฑ)

**templates/pdf/assistant-evaluation.html**

ใช้ {{ }} เป็น placeholder ให้ server เติมค่า

```
<!DOCTYPE html>
```

```
<html lang="th">
<head>
    <meta charset="utf-8" />
    <title>แบบประเมินครุพัชวย</title>
    <link rel="stylesheet" href="th-sarabun.css" />

<style>
    @page {
        size: A4;
        margin: 2.5cm 2cm 2cm 3cm; /* ช้ายเพื่อเข้าแฟ้ม */
    }

    body {
        color: #000;
    }

    .center { text-align: center; }
    .right { text-align: right; }
    .bold { font-weight: bold; }

    .header img {
        width: 90px;
        display: block;
        margin: 0 auto 8px auto;
    }

    .title {
        font-size: 20pt;
        font-weight: bold;
        margin-bottom: 12px;
    }

```

```
}
```

```
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin-top: 8px;  
}
```

```
table, th, td {  
    border: 1px solid #000;  
}
```

```
th, td {  
    padding: 6px;  
    vertical-align: top;  
}
```

```
.no-border {  
    border: none !important;  
}
```

```
.section {  
    margin-top: 16px;  
}
```

```
.checkbox {  
    font-family: monospace;  
    font-size: 14pt;  
}
```

```

.signature {
    height: 60px;
}

</style>

</head>

<body>

<!-- HEADER -->
<div class="header center">
    
    <div class="title">
        แบบบันทึกการประเมินการเตรียมความพร้อม<br/>
        และพัฒนาอย่างเข้ม ตำแหน่งครูผู้ช่วย
    </div>
    <div>
        <input type="checkbox"/> ครั้งที่ 1 &nbsp;&nbsp; <input type="checkbox"/> ครั้งที่ 2
    </div>
</div>

<!-- INFO -->
<table class="no-border">
    <tr class="no-border">
        <td class="no-border">ชื่อผู้รับการประเมิน {{TEACHER_NAME}}</td>
        <td class="no-border">ตำแหน่ง ครูผู้ช่วย</td>
    </tr>
    <tr class="no-border">
        <td class="no-border">สถานศึกษา {{SCHOOL_NAME}}</td>
        <td class="no-border">สังกัด {{AREA_NAME}}</td>
    </tr>

```

```

<tr class="no-border">
    <td class="no-border">ปีการศึกษา {{ACADEMIC_YEAR}}</td>
    <td class="no-border">วันที่ประเมิน {{EVAL_DATE}}</td>
</tr>
</table>

```

```

<!-- SECTION 1 -->
<div class="section bold">
    1. ด้านการปฏิบัติงาน (Work Performance)
</div>

```

```

<table>
    <tr>
        <th width="35%">รายการประเมิน</th>
        <th width="40%">พฤติกรรม/หลักฐาน</th>
        <th width="25%">ผล</th>
    </tr>

```

```

{{WP_ROWS}}
</table>

```

```

<!-- SECTION 2 -->
<div class="section bold">
    2. ด้านคุณลักษณะและจรรยาบรรณวิชาชีพ
</div>

```

```

<table>
    <tr>
        <th width="35%">รายการประเมิน</th>
        <th width="40%">พฤติกรรม/หลักฐาน</th>

```

```
<th width="25%>លត</th>
</tr>
```

```
{ET_ROWS}
</table>
```

```
<!-- SUMMARY -->
<div class="section bold">
    ស្តីពូការបរមេន
</div>
```

```
<div>
     ផែន &nbsp;&nbsp;  មិផែន
</div>
```

```
<div class="section">
    <b>ចុះឈើន</b><br/>
    {{STRENGTHS}}
</div>
```

```
<div class="section">
    <b>ចុះថីគរព័ណាន</b><br/>
    {{IMPROVEMENTS}}
</div>
```

```
<div class="section">
    <b>ខ្លួននេននេន</b><br/>
    {{REMARK}}
</div>
```

```

<!-- SIGNATURES -->
<div class="section">
  <table>
    <tr>
      <td class="center signature">
        ลงชื่อ .....<br/>
        {{EVAL1_NAME}}<br/>
        {{EVAL1_ROLE}}
      </td>
      <td class="center signature">
        ลงชื่อ .....<br/>
        {{EVAL2_NAME}}<br/>
        {{EVAL2_ROLE}}
      </td>
    </tr>
  </table>
</div>

</body>
</html>

```

### **3** Generate PDF ด้วย Puppeteer

src/lib/pdf/generateAssistantEvaluationPDF.ts

```

import puppeteer from "puppeteer";
import fs from "fs";
import path from "path";

export async function generateAssistantEvaluationPDF(html: string) {
  const browser = await puppeteer.launch({
    headless: "new",
  
```

```

args: ["--no-sandbox", "--disable-setuid-sandbox"],
});

const page = await browser.newPage();
await page.setContent(html, { waitUntil: "networkidle0" });

const pdf = await page.pdf({
  format: "A4",
  printBackground: true,
});

await browser.close();
return pdf;
}

```

#### วิธีใช้งานใน API (ตัวอย่าง)

```

const htmlTemplate = fs.readFileSync(
  path.join(process.cwd(), "templates/pdf/assistant-evaluation.html"),
  "utf8"
);

const filledHtml = htmlTemplate
  .replace("{TEACHER_NAME}", teacher.fullName)
  .replace("{SCHOOL_NAME}", school.schoolName)
  .replace("{AREA_NAME}", area.name)
  .replace("{ACADEMIC_YEAR}", year.name)
  .replace("{EVAL_DATE}", "15 มกราคม 2569")
  // เติม WP_ROWS / ET_ROWS / signatures ต่อจาก data จริง
;

```

```
const pdfBuffer = await generateAssistantEvaluationPDF(filledHtml);
```

---

### ✓ จุดเด่นของแนวนี้ (สำคัญมาก)

- ✓ หน้าตา = เอกสารราชการจริง (ครุ/ผล.ไม่ต่อต้าน)
  - ✓ แก้แบบฟอร์มง่าย (HTML/CSS)
  - ✓ รองรับตราครุฑ / ลายเซ็น / ช่องตีก
  - ✓ ต่อ PDF export อื่น ๆ ได้ (รายงานเขต / สรุปภาพรวม)
- 

### ถัดไปครุฑใหญ่ต้องการ

ผู้ทำต่อได้ทันที:

1 พิ่งก์ชัน map data → WP\_ROWS / ET\_ROWS อัตโนมัติ

2 PDF “สรุปสถานะครุฑช่วยระดับโรงเรียน/เขต”

3 เพิ่มลายเซ็นจริง (upload image → embed PDF)

เลือกต่อข้อไหนก่อน บอกผู้ได้เลยครับ 🤗

ทำทั้ง 3 ข้อเลย 1 พิ่งก์ชัน map data → WP\_ROWS / ET\_ROWS อัตโนมัติ 2 PDF “สรุปสถานะครุ

ผู้ช่วยระดับโรงเรียน/เขต” 3 เพิ่มลายเซ็นจริง (upload image → embed PDF)

Thought for 10s

ได้ครับครุฑใหญ่ ✓ ผู้ทำ “ครบทั้ง 3” แบบเอาไปใช้ต่อได้เลย (โค้ด + template + endpoint แนวทางเก็บไฟล์/ลายเซ็น) — ทั้งหมดออกแบบให้ทำงานกับ Prisma schema ที่เราทำไว้ และแนบ “เพิ่ม field เล็กน้อย” เพื่อเก็บลายเซ็นอย่างเป็นทางการ

---

### (0) เพิ่มฟิลด์สำหรับ “ลายเซ็นจริง” ใน DB (แนะนำให้ทำ)

ใน schema.prisma ให้เพิ่ม 2 field ใน AssistantEvaluator:

```
model AssistantEvaluator {
```

```
    id          String @id @default(uuid())
```

```
    evaluationId String
```

```
    evaluation   AssistantEvaluation @relation(fields: [evaluationId], references: [id])
```

```
evaluatorUserId String  
evaluator     User @relation(fields: [evaluatorUserId], references: [id])
```

```
roleTitle   String?  
signedAt    DateTime?
```

// เพิ่ม 2 ฟิลด์นี้  
signatureImageUrl String?  
signatureName String? // ชื่อผู้ลงนามในเอกสาร (เพื่อไม่ตรง displayName)

```
@@unique([evaluationId, evaluatorUserId])  
}
```

แล้ว migrate ตามปกติ (prisma migrate dev)

---

## 1 พังก์ชัน map data → WP\_ROWS / ET\_ROWS อัตโนมัติ

### 1.1 src/lib/pdf/row-mapper.ts

หน้าที่: รับ indicators + rules + evidences + evaluation และสร้าง <tr>...</tr> สำหรับ WP/ET ให้เอง

```
// src/lib/pdf/row-mapper.ts
```

```
import type { EvidenceType, KKSIndicatorDomain } from "@prisma/client";
```

```
type IndicatorWithRules = {  
  id: string;  
  code: string;  
  domain: KKSIndicatorDomain;  
  title: string;  
  description?: string | null;  
  evidenceRules: Array<{  
    evidenceType: EvidenceType;  
    minCount: number;  
    isMandatory: boolean;
```

```
    note: string | null;  
  }>;  
};  
  
type EvidenceLite = {  
  indicatorId: string;  
  type: EvidenceType;  
  title: string;  
  fileUrl: string;  
};  
  
type BuildRowsInput = {  
  indicators: IndicatorWithRules[];  
  evidences: EvidenceLite[];  
  // optional: highlight pass/fail state  
  domainPassed?: {  
    workPerformancePassed?: boolean;  
    ethicsPassed?: boolean;  
  };  
};  
  
function escapeHtml(s: string) {  
  return s  
    .replaceAll("&", "&")  
    .replaceAll("<", "<")  
    .replaceAll(">", ">")  
    .replaceAll("'", """)  
    .replaceAll("'", "'");  
}  
}
```

```

function checkbox(checked: boolean) {
    // ใช้ glyph แบบราชการ: ✓ / □
    return checked ? "✓" : "□";
}

function evidenceLabel(type: EvidenceType) {
    // ตั้งชื่อแบบราชการอ่านง่าย (ปรับได้)
    switch (type) {
        case "LESSON_PLAN": return "แผนการสอน";
        case "TEACHING_MEDIA": return "สื่อ/ใบงาน";
        case "TEACHING_REFLECTION": return "บันทึกหลังสอน/สะท้อนคิด";
        case "ACTIVITY_PHOTO": return "ภาพกิจกรรม";
        case "OFFICIAL_DOCUMENT": return "เอกสารทางราชการ/คำสั่ง/รายงาน";
        default: return "หลักฐานอื่น ๆ";
    }
}

function buildRuleLine(rule: IndicatorWithRules["evidenceRules"][number], have: number) {
    const base = `${evidenceLabel(rule.evidenceType)} อย่างน้อย ${rule.minCount} ชิ้น (มี ${have})`;
    const note = rule.note ? ` - ${rule.note}` : "";
    return `${base}${note}`;
}

function groupEvidence(evidences: EvidenceLite[]) {
    const map = new Map<string, EvidenceLite[]>(); // key indicatorId:type
    for (const e of evidences) {
        const k = `${e.indicatorId}:${e.type}`;
        if (!map.has(k)) map.set(k, []);
        map.get(k)!.push(e);
    }
}

```

```

return map;
}

export function buildIndicatorRowsHTML(input: BuildRowsInput) {
  const evMap = groupEvidence(input.evidences);

  const byDomain = (domain: KKSIndicatorDomain) =>
    input.indicators
      .filter(i => i.domain === domain)
      .sort((a, b) => a.code.localeCompare(b.code));

  const buildRows = (domain: KKSIndicatorDomain) => {
    const list = byDomain(domain);
    const rows: string[] = [];

    for (const ind of list) {
      const mandatoryRules = ind.evidenceRules.filter(r => r.isMandatory);
      const optionalRules = ind.evidenceRules.filter(r => !r.isMandatory);

      // ស្រួលគ្រប់ខ្លួនពាំង indicator (ផែនការ mandatory)
      let indicatorOk = true;
      const lines: string[] = [];

      for (const r of mandatoryRules) {
        const key = `${ind.id}:${r.evidenceType}`;
        const have = evMap.get(key)?.length ?? 0;
        if (have < r.minCount) indicatorOk = false;
        lines.push(`{$checkbox(have >= r.minCount)} ${escapeHtml(buildRuleLine(r, have))}`);
      }
    }
  }
}

```

```

// optional (แสดงเป็น "แนะนำ" ไม่ผูก pass/fail)
for (const r of optionalRules) {
    const key = `${ind.id}:${r.evidenceType}`;
    const have = evMap.get(key)?.length ?? 0;
    lines.push(` ${checkbox(have >= r.minCount)} (แนะนำ) ${escapeHtml(buildRuleLine(r, have))}`);
}

// แนบรายการไฟล์ (ย่อ) — ไม่ส่งยาวเกินในฟอร์มราชการ
const attached: string[] = [];
for (const r of ind.evidenceRules) {
    const key = `${ind.id}:${r.evidenceType}`;
    const arr = evMap.get(key) ?? [];
    // เอาแค่ 2 ข้อ/ประเภทพอ (กันยาว)
    for (const ev of arr.slice(0, 2)) {
        attached.push(`- ${escapeHtml(ev.title)}`);
    }
}

const evidenceBlock = `
<div>
<div>${lines.map(x => `<div>${x}</div>`).join("")}</div>
${attached.length ? `<div style="margin-top:6px"><b>รายการหลักฐาน (ย่อ):</b><br/>${attached.join("<br/>")}</div>` : ""}
</div>
`.trim();

const resultCell = `
<div class="checkbox">${checkbox(indicatorOk)} ฝ่านັ້ນຕໍ່າ</div>
<div class="checkbox">${checkbox(!indicatorOk)} ຍັງໄມ້ຮັບ</div>

```

```

    `trim();

rows.push(`

<tr>
<td><b>${escapeHtml(ind.code)} ${escapeHtml(ind.title)}</b></td>
<td>${evidenceBlock}</td>
<td class="center">${resultCell}</td>
</tr>
`);

}

return rows.join("\n");
};

const wpRows = buildRows("WORK_PERFORMANCE");
const etRows = buildRows("ETHICS");

return { wpRows, etRows };
}

```

## 1.2 การใช้ใน generator

ในตอน fill template แค่เรียก:

```

const { wpRows, etRows } = buildIndicatorRowsHTML({ indicators, evidences });

html = html.replace("{{WP_ROWS}}", wpRows).replace("{{ET_ROWS}}", etRows);

```

## 2 PDF “สรุปสถานะครุภัช่วยระดับโรงเรียน/เขต”

### 2.1 Template: templates/pdf/assistant-status-summary.html

เป็นเอกสารราชการสรุปภาพรวม (A4 / TH Sarabun / มีตราครุฑ) พร้อมตาราง “รายชื่อ + ความพร้อม + ขาดอัตรา”

```
<!DOCTYPE html>
```

```
<html lang="th">
<head>
    <meta charset="utf-8" />
    <title>សរុបសាននភ្លើងម៉ាយ</title>
    <link rel="stylesheet" href="th-sarabun.css" />
    <style>
        @page { size: A4; margin: 2.5cm 2cm 2cm 3cm; }
        body { color:#000; }
        .center { text-align:center; }
        .right { text-align:right; }
        .bold { font-weight:bold; }
        .header img { width: 90px; display:block; margin:0 auto 8px auto; }
        .title { font-size: 20pt; font-weight:bold; margin-bottom: 8px; }
        table { width:100%; border-collapse:collapse; margin-top:10px; }
        th, td { border:1px solid #000; padding:6px; vertical-align:top; }
        .tag-ok { font-weight:bold; }
        .tag-bad { font-weight:bold; }
        .small { font-size: 14pt; }
    </style>
</head>

<body>
    <div class="header center">
        
        <div class="title">រាយការណាស់សាននភ្លើងម៉ាយ</div>
        <div class="bold">{{SCOPE_TITLE}}</div>
        <div>បីវិទ្យាឌភី {{ACADEMIC_YEAR}} | រំលែក {{ROUND}}</div>
        <div class="small">ចំណាំ នៃ ថ្ងៃ {{REPORT_DATE}}</div>
    </div>
```

```

<table>
  <tr>
    <th width="7%">ลำดับ</th>
    <th width="23%">ชื่อ-สกุล</th>
    <th width="25%">สถานศึกษา</th>
    <th width="15%">สถานะ</th>
    <th width="30%">ขาดหลักฐานขึ้นตໍา (สรุป)</th>
  </tr>
  {{ROWS}}
</table>

<div style="margin-top:16px" class="small">
  <b>หมายเหตุ:</b> รายการ “ขาดหลักฐานขึ้นตໍา” คำนวณจากติKA (KKSIndicatorEvidenceRequired)
  ในระบบ
</div>

<div style="margin-top:20px">
  <table>
    <tr>
      <td class="center" style="height:80px">
        ลงชื่อ .....<br/>
        {{REPORT_SIGN_NAME}}<br/>
        {{REPORT_SIGN_ROLE}}
      </td>
    </tr>
  </table>
</div>
</body>
</html>

```

## 2.2 Builder: src/lib/pdf/summary-mapper.ts

```

// src/lib/pdf/summary-mapper.ts

function escapeHtml(s: string) {
    return s
        .replaceAll("&", "&")
        .replaceAll("<", "<")
        .replaceAll(">", ">")
        .replaceAll("'", """)
        .replaceAll("'', "&#039;");
}

export type SummaryRow = {
    teacherName: string;
    schoolName: string;
    overallReady: boolean;
    missingSummary: string; // e.g., "WP.1: TEACHING_MEDIA; ET.2: TEACHING_REFLECTION"
};

export function buildSummaryRowsHTML(rows: SummaryRow[]) {
    return rows
        .map((r, idx) => {
            const status = r.overallReady
                ? `<span class="tag-ok">พร้อมประเมิน</span>`
                : `<span class="tag-bad">ยังไม่พร้อม</span>`;

            return `
<tr>
<td class="center">${idx + 1}</td>
<td>${escapeHtml(r.teacherName)}</td>
<td>${escapeHtml(r.schoolName)}</td>
<td class="center">${status}</td>

```

```

<td>${escapeHtml(r.missingSummary || "-")}</td>
</tr>
`.trim();
})
.join("\n");
}

```

---

### 3 เพิ่มลายเซ็นจริง (upload image → embed PDF)

แนวทาง “ถูกต้องและปลอดภัย”:

1. ให้ผู้ลงนาม upload รูปลายเซ็นขึ้น **Supabase Storage** จากฝั่ง client
2. ได้ public URL หรือ signed URL (ถ้า bucket private)
3. ส่ง URL เข้า API เพื่อบันทึกลง AssistantEvaluator.signatureImageUrl
4. เวลา generate PDF → embed  ลงตำแหน่งลายเซ็น

#### 3.1 ฝั่ง DB: บันทึกลายเซ็น

**Endpoint: POST /api/assistant-evaluations/:id/signature**

app/api/assistant-evaluations/[evaluationId]/signature/route.ts

// app/api/assistant-evaluations/[evaluationId]/signature/route.ts

```
import { prisma } from "@/lib/prisma";
```

```
import { getActorFromRequest } from "@/lib/auth";
```

```
import { rbacErrorToResponse, RBACError } from "@/lib/rbac";
```

```
export async function POST(req: Request, ctx: { params: { evaluationId: string } }) {
```

```
try {
```

```
    const actor = await getActorFromRequest(req);
```

```
    const evaluationId = ctx.params.evaluationId;
```

```
    const body = await req.json();
```

```
    const signatureImageUrl: string = body.signatureImageUrl;
```

```
    const signatureName: string | undefined = body.signatureName;
```

```

if (!signatureImageUrl) throw new RBACError("BAD_REQUEST", "signatureImageUrl required");

// ต้องเป็น "กรรมการใน evaluation นี้" เท่านั้นถึงจะลงนามได้
const evaluator = await prisma.assistantEvaluator.findFirst({
  where: { evaluationId, evaluatorUserId: actor.id },
  select: { id: true },
});

if (!evaluator) throw new RBACError("FORBIDDEN", "You are not an evaluator for this
evaluation");

const updated = await prisma.assistantEvaluator.update({
  where: { id: evaluator.id },
  data: {
    signatureImageUrl,
    signatureName: signatureName ?? undefined,
    signedAt: new Date(),
  },
});

return Response.json({ data: updated, meta: {} }, { status: 200 });
} catch (e) {
  return rbacErrorToResponse(e);
}
}

```

### 3.2 ฝั่ง Client: Upload ไฟล์ Supabase Storage (ตัวอย่าง)

สมมติใช้ bucket signatures (ตั้ง public หรือ private ก็ได้)

```

// client example (tsx)
import { createClient } from "@supabase/supabase-js";

```

```

const supabase = createClient(process.env.NEXT_PUBLIC_SUPABASE_URL!,
process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!);

export async function uploadSignature(file: File, userId: string) {
  const ext = file.name.split(".").pop() || "png";
  const path = `signatures/${userId}/${Date.now()}.${ext}`;

  const { error } = await supabase.storage.from("signatures").upload(path, file, {
    cacheControl: "3600",
    upsert: true,
    contentType: file.type,
  });
  if (error) throw error;

  // ถ้า bucket public:
  const { data } = supabase.storage.from("signatures").getPublicUrl(path);
  return data.publicUrl;
}

ถ้า bucket private: ใช้ createSignedUrl และเอา signed URL ไป embed ตอน generate PDF (หรือจะ
generate pdf ทันทีหลัง signed ก็ได้)

```

#### 4) ใส่ “ลายเซ็นจริง” ลง PDF Template แบบประเมิน

ปรับใน assistant-evaluation.html ตรงช่องลายเซ็นให้รองรับรูป:

```

<td class="center signature">
  {{EVAL1_SIGNATURE_IMG}}
  ลงชื่อ .....<br/>
  ({{EVAL1_NAME}})<br/>
  {{EVAL1_ROLE}}
</td>

```

แล้วเวลา fill:

- ถ้ามีลายเซ็น → ใส่ 
  - ถ้าไม่มี → ใส่ค่าว่าง
- 

## 5) Generator รวม: ประเมิน (WP/ET auto rows) + signatures + PDF

**src/lib/pdf/buildAssistantEvaluationHtml.ts**

```
import fs from "fs";
import path from "path";
import { buildIndicatorRowsHTML } from "@lib/pdf/row-mapper";

type BuildEvalHtmlArgs = {
  templatePath: string;
  garudaBase64DataUrl: string;

  teacherName: string;
  schoolName: string;
  areaName: string;
  academicYear: string;
  evalDateText: string;

  roundText: "ครึ่งที่ 1" | "ครึ่งที่ 2";
  passFail: "PASS" | "FAIL";

  strengths?: string;
  improvements?: string;
  remark?: string;

  indicators: any[];
  evidences: any[];
}
```

```

evaluators: Array<{
  name: string;
  roleTitle: string;
  signatureImageUrl?: string | null;
}>;
};

export function buildAssistantEvaluationHtml(args: BuildEvalHtmlArgs) {
  const htmlTemplate = fs.readFileSync(path.join(process.cwd(), args.templatePath), "utf8");

  const { wpRows, etRows } = buildIndicatorRowsHTML({
    indicators: args.indicators,
    evidences: args.evidences,
  });

  const eval1 = args.evalutors[0] ?? { name: "-", roleTitle: "-", signatureImageUrl: null };
  const eval2 = args.evalutors[1] ?? { name: "-", roleTitle: "-", signatureImageUrl: null };

  const sigImg = (url?: string | null) =>
    url ? `` : "";

  let html = htmlTemplate
    .replace("{{GARUDA_BASE64}}", args.garudaBase64DataUrl)
    .replace("{{TEACHER_NAME}}", args.teacherName)
    .replace("{{SCHOOL_NAME}}", args.schoolName)
    .replace("{{AREA_NAME}}", args.areaName)
    .replace("{{ACADEMIC_YEAR}}", args.academicYear)
    .replace("{{EVAL_DATE}}", args.evalDateText)
    .replace("{{WP_ROWS}}", wpRows)
    .replace("{{ET_ROWS}}", etRows)
}

```

```

.replace("{{STRENGTHS}}", args.strengths ?? "-")
.replace("{{IMPROVEMENTS}}", args.improvements ?? "-")
.replace("{{REMARK}}", args.remark ?? "-")
.replace("{{EVAL1_NAME}}", eval1.name)
.replace("{{EVAL1_ROLE}}", eval1.roleTitle)
.replace("{{EVAL2_NAME}}", eval2.name)
.replace("{{EVAL2_ROLE}}", eval2.roleTitle);

// แทรกรูปลายเซ็นถ้ามี
html = html
.replace("{{EVAL1_SIGNATURE_IMG}}", sigImg(eval1.signatureImageUrl))
.replace("{{EVAL2_SIGNATURE_IMG}}", sigImg(eval2.signatureImageUrl));

// ตີ່ກຽບຮັ້ງທີ 1/2 ແລະ ຜ່ານ/ໄມ່ຜ່ານ (ແບບຈ່າຍ)
html = html.replace("□ ຄຽບຮັ້ງທີ 1", args.roundText === "ຄຽບຮັ້ງທີ 1" ? "☑ ຄຽບຮັ້ງທີ 1" : "□ ຄຽບຮັ້ງທີ 1");
html = html.replace("□ ຄຽບຮັ້ງທີ 2", args.roundText === "ຄຽບຮັ້ງທີ 2" ? "☑ ຄຽບຮັ້ງທີ 2" : "□ ຄຽບຮັ້ງທີ 2");
html = html.replace("□ ຜ່ານ", args.passFail === "PASS" ? "☑ ຜ່ານ" : "□ ຜ່ານ");
html = html.replace("□ ໄມ່ຜ່ານ", args.passFail === "FAIL" ? "☑ ໄມ່ຜ່ານ" : "□ ໄມ່ຜ່ານ");

return html;
}

```

## 6) Generator รวม: PDF ສຽບສຕານະຄຽບຜູ້ຂ່າຍ (ໂຮງເຮືອນ/ເຂດ)

`src/lib/pdf/buildAssistantSummaryHtml.ts`

```

import fs from "fs";
import path from "path";
import { buildSummaryRowsHTML, SummaryRow } from "@/lib/pdf/summary-mapper";

export function buildAssistantSummaryHtml(args: {

```

```
templatePath: string;
garudaBase64DataUrl: string;
scopeTitle: string; // "ຮະດັບໂຮງເຮືຍນ..." / "ຮະດັບເຂດ..."
academicYear: string;
round: string; // ROUND_1/ROUND_2
reportDateText: string;
signerName: string;
signerRole: string;
rows: SummaryRow[];
}) {
const tpl = fs.readFileSync(path.join(process.cwd(), args.templatePath), "utf8");
const rowsHtml = buildSummaryRowsHTML(args.rows);

return tpl
.replace("{{GARUDA_BASE64}}", args.garudaBase64DataUrl)
.replace("{{SCOPE_TITLE}}", args.scopeTitle)
.replace("{{ACADEMIC_YEAR}}", args.academicYear)
.replace("{{ROUND}}", args.round)
.replace("{{REPORT_DATE}}", args.reportDateText)
.replace("{{ROWS}}", rowsHtml)
.replace("{{REPORT_SIGN_NAME}}", args.signerName)
.replace("{{REPORT_SIGN_ROLE}}", args.signerRole);
}
```