

Requirement

การลงทะเบียนผู้ปฏิบัติงาน จะเริ่มจาก

- superuser หรือ Admin ซึ่งจะมีสิทธิ์เต็ม เริ่มจาก เข้าหน้าบริหารจัดการสำหรับ Admin ฝั่ง Frontend ไม่ใช่ backend ของ Django เข้าไปเพิ่มหน่วยงานต่างๆ
- บทบาทมี Admin สิทธิ์เต็มที่ officer สิทธิ์เต็ม เข้าได้ทุกหน่วยงาน ยกเว้นฟีเจอร์ตั้งค่า submitter สิทธิ์เต็ม เฉพาะในหน่วยงานตัวเอง
- หน่วยงานจะมีลำดับ(จำเป็น) ชื่อเต็มหน่วยงาน(จำเป็น) ชื่อย่อหน่วยงาน คำอธิบาย Admin CRUD ได้หมด ยกเว้น primary key
- เมื่อสร้างหน่วยงานแล้ว Admin ก็จะเป็นการสร้าง submitter ของแต่ละหน่วยงาน เพื่อเข้าระบบมาทำการอัปเดตข้อมูล
- submitter มีลำดับ ชื่อหน่วยงาน ชื่อ-สกุล user pass ของ submitter โครงสร้างแล้วแต่คุณจะทำแบบ
- user pass ของ submitter Admin กำหนดให้ แต่สามารถเปลี่ยนได้ที่หลัง ชื่อ สกุล pass เปลี่ยนได้หมด ยกเว้น user อันนี้ก็ตามหลักการ สรุป Admin CRUD user ได้
- พัฒนาด้วย django framework และ mysql
- ชื่อระบบ NPU-CeremonyBadge
- ชื่อย่อ (System Code):NPU-CeremonyBadge
- ระบบออกบัตรสำหรับผู้ปฏิบัติงานพิธีพระราชทานปริญญาบัตร มหาวิทยาลัยนครพนม ประจำปี 2567 (Nakhon Phanom University Ceremony Staff Badge Issuing System 2024)
- เมื่อได้ หน่วยงาน ได้ submitter แล้ว submitter login เข้ามากรอกข้อมูลของหน่วยงานตัวเอง
- ความต้องการ ทำ On-demand printing + แก้วใส (ผลงานเตรียมล่วงหน้า, ปรับได้ทันที) อันนี้ทำ CRUD ให้ submitter อยู่แล้ว

- สรุปแนวสี (Theme Guideline) Theme สีพาสเทลเรียบราชการ (ม่วง-ขาว-ดำ) จะให้ความรู้สึก “เป็นระบบ พิธี แต่ทันสมัย” และเข้ากับภาพลักษณ์ของมหาวิทยาลัยได้ดีมาก

หมวด	สีหลัก / โค้ด	การใช้งาน
Primary (ม่วงพาสเทล)	#A78BFA (ม่วงลาเวนเดอร์) / #7C3AED (ม่วงเข้ม) ตัวอักษร/หัวข้อ	Navbar, ปุ่มหลัก, หัวข้อ, Highlight

หมวด	สีหลัก / โค้ด	การใช้งาน
Secondary (เทาอ่อน / ม่วงเทา)	#E9D8FD (พื้นปุ่มรอง / hover), #D1C4E9	พื้นหลังการ์ด, ปุ่มรอง
Accent - Success (เขียวพาสเทล)	#A7F3D0 / #10B981	ผ่าน, อนุมัติ, สถานะ “ใช้งานได้”
Accent - Warning (ทองพาสเทล)	#FDE68A / #F59E0B	เตือน, ยังไม่ตรวจสอบ
Accent - Danger (ชมพูพาสเทล)	#FECACA / #EF4444	ปฏิเสธ, หมดสิทธิ์
พื้นหลัง (Background)	#FFFFFF	พื้นหลักทั้งหมด
ตัวหนังสือ (Text)	#212529	สีมาตรฐาน อ่านง่ายชัด
เส้นคั่น / เส้นขอบ (Border)	#E5E7EB	ใช้ในตาราง / การ์ด / ช่องกรอก

ปุ่ม (Buttons)

แบบที่แนะนำ:

- ☐ ใช้ Outline Buttons สำหรับโหมตราชการ / ขาวสะอาด (ช่วยลดความหนาแน่นของสี)
- ☐ ใช้ Solid Buttons (bg สีเต็ม) เฉพาะจุดสำคัญ เช่น ปุ่ม “บันทึก”, “พิมพ์บัตร”, “อนุมัติ”

<!-- ปุ่มหลัก -->

```
<button type="button" class="btn btn-outline-primary">บันทึกข้อมูล</button>
```

<!-- ปุ่มเตือน -->

```
<button type="button" class="btn btn-outline-warning">รอตรวจสอบ</button>
```

<!-- ปุ่มผ่าน -->

```
<button type="button" class="btn btn-outline-success">อนุมัติ</button>
```

<!-- ปุ่มอันตราย -->

```
<button type="button" class="btn btn-outline-danger">ยกเลิกบัตร</button>
```

<!-- ปุ่มพิเศษ (Solid) -->

```
<button type="button" class="btn btn-primary text-white fw-bold shadow-sm">
```

พิมพ์บัตร

ตัวอย่างรวมสี่ + ปุ่มบนพื้นหลังขาว

ตัวอย่าง

โค้ด

ปุ่มแบบ Outline pastel บนพื้นขาว ดูเบาแต่เด่น

ปุ่ม Solid สำหรับ action หลัก เช่น “บันทึก”, “พิมพ์บัตร”

แนะนำโครง Navbar / Dashboard

```
<nav class="navbar navbar-expand-lg navbar-light bg-white border-bottom shadow-sm">
  <a class="navbar-brand fw-bold text-primary ms-3" href="#"><img alt="NPU Ceremony Badge" data-bbox="688 228 718 268"/> NPU CeremonyBadge</a>
  <div class="ms-auto me-3">
    <button class="btn btn-outline-dark btn-sm">ออกจากระบบ</button>
  </div>
</nav>
```






```
<div class="container py-4">
  <div class="row g-3">
    <div class="col-md-3">
      <div class="card border-0 shadow-sm p-3">
        <h6 class="text-secondary">บัตรทั้งหมด</h6>
        <h2 class="fw-bold text-primary">2,356</h2>
      </div>
    </div>
  </div>
</div>
```

ข้อเสนอสำหรับ “NPU-CeremonyBadge”

✓ ใช้แบบ A: Sidemenu + Dashboard

เป็นมาตรฐานของระบบบริหารงานภายใน เช่น Smart Office, e-Document, ระบบอบรม, และระบบออกบัตร
พิธีในมหาวิทยาลัยอื่น ๆ

ตัวอย่างเมนูฝั่งซ้าย (Sidebar)

- ☐  Dashboard
- ☐  รายชื่อผู้ปฏิบัติงาน
- ☐  ออกบัตร / พิมพ์บัตร
- ☐  รายงานสรุป
- ☐  ตั้งค่า / Template บัตร

บนสุด (Top Bar)

- ☐ โลโก้ + ชื่อระบบ
- ☐ ปุ่ม “ออกจากระบบ”
- ☐ ชื่อผู้ใช้ / บทบาท
- ☐ (อาจมีปุ่มลัด: พิมพ์บัตร, ตรวจสอบบัตร, สแกน)

เป็น Responsive Design

บทบาทผู้ใช้และสิ่งที่แต่ละกลุ่ม “ควรเห็นในระบบ”

1. Submitter (ผู้กรอกข้อมูลในหน่วยงาน)

เป็นผู้แทนของแต่ละหน่วยงาน ที่กรอกข้อมูลบุคลากรเข้าร่วมพิธี

หน้าหลักหลัง Login: “ภาพรวมของหน่วยงานตนเอง”

- ☐ จำนวนบุคลากรที่ส่งข้อมูลแล้ว / ทั้งหมด
- ☐ สถานะการตรวจสอบแต่ละคน (✓ ผ่าน, ☐ รอตรวจ, ✗ ส่งกลับแก้ไข)
- ☐ ปุ่มเพิ่ม/แก้ไขข้อมูล (เช่น แก้รูป, เพิ่มรายชื่อ)
- ☐ ปุ่ม “ส่งข้อมูลให้ตรวจ” (submit รอบสุดท้าย)
- ☐ export รายชื่อของหน่วยงาน (เฉพาะของตัวเอง)

ตารางข้อมูลอยู่ในหน้าเดียวกัน

ชื่อ	ประเภทบัตร	สถานะ	ตรวจแล้วโดย	วันที่ส่ง
นาย ก สีชมพู	– บุคคลชั้นในสุด	✓ ผ่าน	officerA	30 ต.ค.
นาง ข สีเหลือง	– บุคคลชั้นกลาง	□ รอตรวจ	-	-

2. Officer (เจ้าหน้าที่ตรวจสอบข้อมูล)

ตรวจสอบข้อมูลแต่ละหน่วยงานก่อนอนุมัติให้พิมพ์บัตร

Dashboard หลัง Login:

- ☐ รายการหน่วยงานที่ส่งข้อมูลเข้ามาแล้ว (พร้อมจำนวนคน)
- ☐ สรุปสถานะการตรวจสอบ เช่น
 - ✓ ผ่านการตรวจทั้งหมด xx หน่วยงาน
 - □ อยู่ระหว่างตรวจ xx หน่วยงาน
 - ✗ ส่งกลับแก้ไข xx หน่วยงาน
- ☐ ปุ่มลัด “เข้าไปตรวจหน่วยงาน”
- ☐ กรองข้อมูลตาม ประเภทบัตร (ชมพู / แดง / เหลือง / เขียว)

ในรายละเอียดแต่ละหน่วยงาน

- ☐ รายชื่อทั้งหมดจาก Submitter
- ☐ แสดงรูปภาพแนบ / ตรวจเอกสารแนบ / แก้สถานะ
- ☐ ปุ่ม “อนุมัติทั้งหมด” หรือ “ส่งกลับให้แก้ไข”

3. Admin (ผู้ดูแลระบบกลาง) เห็นภาพรวมทั้งหมดของพิธี และควบคุมการออกบัตรจริง

Dashboard กลาง (สรุปสูงสุด):

- ☐ จำนวนข้อมูลทั้งหมด (ทุกหน่วยงาน)
 - บัตรชมพู: xx คน
 - บัตรแดง: xx คน
 - บัตรเหลือง: xx คน
 - บัตรเขียว: xx คน
- ☐ จำนวน “อนุมัติแล้ว / ยังไม่ตรวจ / ส่งกลับแก้ไข”

- ☐ จำนวนที่พิมพ์บัตรแล้ว / ยังไม่พิมพ์
- ☐ จำนวนผู้ปฏิบัติงานแต่ละประเภท / หน่วยงาน
- ☐ กราฟแท่งแสดงจำนวนบัตรตามหน่วยงาน
- ☐ กราฟวงกลมแสดงสัดส่วนประเภทบัตร

สิทธิ์เพิ่มเติม

- ☐ ตรวจสอบ / แก้ไขข้อมูลได้ทุกหน่วยงาน
- ☐ พิมพ์บัตร (รวม / เฉพาะกลุ่ม)
- ☐ สร้าง QR Code
- ☐ export ข้อมูลทั้งหมดเป็น Excel/PDF

ตัวอย่างการจัดวาง Dashboard รวม (สำหรับ Admin + Officer)

Card ด้านบน (Summary)




การ์ด	ข้อมูล	สีที่ใช้
 บัตรทั้งหมด	2,450 ใบ	ม่วงพาสเทล
 ตรวจอนุมัติแล้ว	2,100	เขียวพาสเทล
<input type="checkbox"/> รอตรวจสอบ	250	ทองพาสเทล
 ส่งกลับแก้ไข	100	ชมพูพาสเทล

Chart Section (กลางหน้า)

- ☐ กราฟวงกลม (Pie chart): สัดส่วนประเภทบัตร (ชมพู/แดง/เหลือง/เขียว)
- ☐ กราฟแท่ง (Bar chart): จำนวนผู้ปฏิบัติงานแต่ละหน่วยงาน
- ☐ แผนภาพแนวโน้ม: สถานะอนุมัติรายวัน

ตารางล่าง (Latest updates)

หน่วยงาน	ส่งเข้ามา	ตรวจแล้ว	พิมพ์แล้ว	หมายเหตุ
มทบ.210	35	30	25	รอตรวจเพิ่ม

หน่วยงาน ส่งเข้ามา ตรวจสอบแล้ว พิมพ์แล้ว หมายเหตุ

กอ.รมน. 12 12 12 -

กยศ. 20 18 18 ส่งกลับ 2 รายการ

ประเภทของบัตร (สีประจำ)

ประเภท สีบัตร แนวการแสดง

บุคคลชั้นในสุด สีชมพู #FECACA มีรูป, QR Code

บุคคลชั้นใน สีแดง #FCA5A5 มีรูป, QR Code

บุคคลชั้นกลาง สีเหลือง #FDE68A ไม่มีรูป, QR Code

บุคคลชั้นนอก สีเขียว #A7F3D0 ไม่มีรูป, QR Code

สรุปแนวทางออกแบบ (ก่อนทำ UI)

บทบาท หน้าหลักหลัง Login พีเจียร์หลัก

Submitter Dashboard หน่วยงาน เพิ่ม/แก้รายชื่อ, ส่งตรวจ, ดูสถานะ, export

Officer Dashboard ตรวจสอบ ตรวจสอบแต่ละหน่วยงาน, อนุมัติ/ส่งกลับ, ดูภาพรวมสถานะ

Admin Dashboard รวม ดูภาพรวมทั้งระบบ, พิมพ์บัตร, แก้ไข, export, สถิติ

เราจะออกแบบ หน้า “Submitter Wizard” (ขั้นตอนกรอกข้อมูล-อัปโหลดรูป-ตรวจบัตร) โดยยังไม่ลงโค้ดจริง แต่จะวาง กรอบโครงหน้ากับหลักการทำงาน + library ที่เหมาะกับ Django ให้ครบก่อนครับ

หน้าที่ Submitter ใช้งาน: “แบบฟอร์มลงทะเบียนและสร้างบัตร”


Navigator Process 3 ขั้นตอน

ขั้นตอน	หน้าชื่อ	รายละเอียด	สถานะ
Step 1: กรอกข้อมูลส่วนบุคคล	แบบฟอร์มกรอกข้อมูลตามไฟล์ Excel (ชื่อ, นามสกุล, เลข 13 หลัก, หน่วยงาน, ประเภทบุคคล, หน้าที่, อายุ, รายนต์, การรับ มีปุ่ม “ถัดไป” วัคซีน ฯลฯ)		

ขั้นตอน	หน้าชื่อ	รายละเอียด	สถานะ
Step 2: อัปโหลดรูปภาพ	สำหรับบัตรสี่ชมพู / แดง เท่านั้น — อัปโหลดรูปภาพ, หมุน, ครอบ (Crop) ได้ — Preview แบบเรียลไทม์	มีปุ่ม “บันทึก รูป + ถัดไป”	
Step 3: ตรวจสอบก่อนยืนยัน	แสดงภาพตัวอย่างบัตรจริง (Template สีตามประเภทบัตร) พร้อมข้อมูลที่กรอก — หากถูกต้องกด “ยืนยันและส่งข้อมูล”	บันทึกลงฐานข้อมูล	

โครงร่าง UI (Bootstrap 5 + Wizard Pattern)


```

<div class="card border-0 shadow-sm">
  <div class="card-header bg-white border-bottom">
    <ul class="nav nav-pills nav-justified">
      <li class="nav-item"><a href="#step1" class="nav-link active">1 

```

Step 2 : อัปโหลดรูป + หมุน + ครอบ ภาพ

Libraries ยอดนิยมนำมาใช้ Django

ฟังก์ชัน	Library	หมายเหตุ
Upload + Preview	 django-imagekit หรือ django-versatileimagefield	สร้าง thumbnail preview อัตโนมัติ

ฟังก์ชัน	Library	หมายเหตุ
Crop/Rotate บน Browser	✓ Cropper.js (frontend JS)	ใช้ร่วมกับ Bootstrap ได้ดี มาก
ตรวจสอบขนาด/สัดส่วนก่อน submit	✓ cropper.getData() หรือ getCanvasData()	ตรวจสอบ aspect ratio เช่น 4:5 หรือ 1:1
ตรวจสอบ quality/resolution	ใช้ JS เช็ค naturalWidth/Height > 600px ขึ้นไป	

Workflow เบื้องหลัง

1. ผู้ใช้เลือกไฟล์ → preview ขึ้นใน
2. Cropper.js เปิดครอบภาพ → หมุน/ซูมได้
3. เมื่อกด “บันทึก รูป” → ส่งข้อมูล crop (rect x,y,w,h,rotate) ไป backend
4. Django ตัดภาพจริงตามค่านี้ ด้วย Pillow ก่อนเก็บ ใน MEDIA_ROOT/photos/
5. ถ้าความละเอียดต่ำ หรือ เอียงเกิน → แจ้งเตือนก่อน submit

ตัวอย่าง โค้ด สั้น ผัง Template

```
<div class="upload-section text-center">
  <input type="file" id="photoInput" accept="image/*" class="form-control">
  <div class="mt-3">
    <img id="photoPreview" class="img-fluid border rounded shadow-sm" style="max-
height:400px;">
  </div>
  <div class="btn-group mt-2">
    <button type="button" class="btn btn-outline-secondary" id="rotateLeft">↶ หมุนซ้าย
  </button>
    <button type="button" class="btn btn-outline-secondary" id="rotateRight">↷ หมุนขวา
  </button>
    <button type="button" class="btn btn-outline-primary" id="cropSubmit">บันทึกรูปนี้</button>
  </div>
</div>
```

ฝัง JS กับ Cropper.js

```
var cropper;

photoInput.onChange = function(e){
  const file = e.target.files[0];
  const url = URL.createObjectURL(file);
  photoPreview.src = url;
  if(cropper){ cropper.destroy(); }
  cropper = new Cropper(photoPreview, {
    aspectRatio: 4/5,
    viewMode: 1,
    rotatable: true,
  });
};

document.getElementById('rotateLeft').onclick = ()=> cropper.rotate(-90);
document.getElementById('rotateRight').onclick = ()=> cropper.rotate(90);
document.getElementById('cropSubmit').onclick = ()=>{
  const canvas = cropper.getCroppedCanvas({width:600,height:750});
  canvas.toBlob(blob=>{
    // upload blob via FormData / fetch
  });
};
```

Step 3 : Preview บน Template บัตร

เมื่อกด “ถัดไป” จาก Step 2 → ระบบจะแสดงหน้า Preview บัตรตาม ประเภทบุคคล

ประเภท สีพื้นบัตร เงื่อนไขรูป

สีชมพู – บุคคลชั้นในสุด #FECACA มีรูป

สีแดง – บุคคลชั้นใน #FCA5A5 มีรูป

สีเหลือง – บุคคลชั้นกลาง #FDE68A ไม่มีรูป

ประเภท สีพื้นบัตร เงื่อนไขรูป

สีเขียว – บุคคลชั้นนอก #A7F3D0 ไม่มีรูป

Preview อาจใช้ HTML template ง่าย ๆ แสดงรูป + ชื่อ + หน่วยงาน + QR โค้ดจำลอง เช่น

```
<div class="card p-4 text-center" style="background-color:#FECACA; width:350px;">
  
  <h5 class="fw-bold mb-0">พ.ต. สสส นนน</h5>
  <small>มทบ.210 — บุคคลชั้นในสุด</small>
  <div class="mt-2"></div>
</div>
```

สรุปแนวทาง

ส่วน	เทคโนโลยี/Library	หมายเหตุ
Process Navigator	SmartWizard หรือ Bootstrap Tabs + JS	ง่ายต่อการ integrate Django template
อัปโหลด + Preview ภาพ	Cropper.js + django-imagekit	รองรับหมุน/ครอบ/ตรวจสอบขนาด
ประมวลผล ภาพฝั่ง server	Pillow (Python)	resize/crop ตาม x,y,w,h
เก็บภาพ Media	MEDIA_ROOT/photos	ตั้งชื่อไฟล์ตาม staff id
Template บัตร Preview	Bootstrap Card + background สีบัตร	แสดงข้อมูล+รูป

บทบาทผู้ใช้งาน (User Roles) — ฉบับปรับตามจริงของระบบ NPU-CeremonyBadge

บทบาท	หน้าที่หลัก	สิทธิ์การเข้าถึง / การใช้งาน	หมายเหตุ
ตัวแทนของ	หน่วยงาน / กรอก	- เข้าถึงเฉพาะข้อมูลของหน่วยงานตนเอง- กรอก / แก้ไข / ลบข้อมูลได้เฉพาะในหน่วยงาน- อัปโหลด	ใช้ workflow 3 ขั้นตอน: กรอก →

บทบาท	หน้าที่หลัก	สิทธิ์การเข้าถึง / การใช้งาน	หมายเหตุ
	ข้อมูลบุคลากรเพื่อ ขอออกบัตร	รูปภาพ / ตรวจสอบ preview / ส่งข้อมูลขออนุมัติ- ดู สถานะการอนุมัติ / พิมพ์รายการของหน่วยงานได้	อัปโหลดรูป → ตรวจสอบและส่ง
Officer	เจ้าหน้าที่ ผู้รับผิดชอบ ตรวจสอบและ จัดการข้อมูลบัตร	- เข้าถึงทุกฟังก์ชันของระบบ ยกเว้น System Setting- ดู/แก้ไข/อนุมัติ/พิมพ์บัตรของทุกหน่วยงาน- จัดการเทม เพลตบัตร (ย่อ), export รายงาน, ตรวจสอบ สถานะการพิมพ์- แก้ไขข้อมูลผิดพลาดหรือส่งกลับให้ หน่วยงานได้- ดู Dashboard รวม (สถิติทั้งหมด)	มีสิทธิ์เทียบเท่า Admin ด้านการ ดำเนินงาน
Admin	ผู้ดูแลระบบหลัก	- เข้าถึงทุกฟังก์ชันทั้งหมด- จัดการผู้ใช้ / บทบาท / สิทธิ์- ตั้งค่าระบบ (System Setting)- กำหนดค่าคงที่ เช่น สีบัตร, template, คีย์ QR, export path- Backup / Restore ข้อมูล	เป็น Superuser ของ ระบบ

สรุปความสัมพันธ์สิทธิ์

☐ Officer = Admin - (System Setting Permission)

☐ Submitter มีเฉพาะส่วนของหน่วยงานตัวเอง

Workflow (ตามสิทธิ์ใหม่)

ลำดับขั้นตอน	Submitter	Officer	Admin
กรอกข้อมูล / อัปโหลดรูป	✓	✓ (แก้ไขทั้งหมด)	✓
ตรวจสอบ / อนุมัติ	✗	✓	✓
สร้างบัตร / พิมพ์บัตร	✗	✓	✓
จัดการเทมเพลตบัตร	✗	✓	✓
Export / รายงาน	✓ (เฉพาะหน่วยงานตนเอง)	✓	✓
ตั้งค่าระบบ (System Setting)	✗	✗	✓

บทบาท : Submitter

สำหรับหน่วยงานผู้กรอกข้อมูล

เมนูหลัก

1. Dashboard (ภาพรวมหน่วยงาน)
 - จำนวนบุคลากรทั้งหมด / ส่งแล้ว / รอตตรวจ
 - สถานะการอนุมัติแต่ละคน
 - ปุ่ม “เพิ่มข้อมูลใหม่” / “ส่งข้อมูลทั้งหมด”
2. แบบฟอร์มบุคลากร
 - Step 1 : กรอกข้อมูล
 - Step 2 : อัปโหลดรูป (หมุน / Crop / Preview)
 - Step 3 : ตรวจสอบก่อนส่ง
 - บันทึกเป็น draft / submit
3. รายการบุคลากรของหน่วยงาน
 - ตารางแสดงชื่อ-ประเภทบัตร-สถานะ
 - ค้นหา / แก้ไข / ลบ / ส่งซ้ำ
4. ส่งข้อมูลให้ตรวจสอบ
 - สรุปรายชื่อที่จะส่ง
 - กด “ยืนยันส่ง” → เปลี่ยนสถานะเป็น Submitted
5. รายงานของหน่วยงาน
 - ดูรายการที่ได้รับอนุมัติแล้ว
 - พิมพ์รายชื่อ / export เป็น Excel เฉพาะหน่วยงาน

บทบาท : Officer สิทธิ์เท่ากับ Admin ยกเว้น System Setting

เมนูหลัก

1. Dashboard รวม
 - สรุปจำนวนบัตรตามประเภท (ชมพู/แดง/เหลือง/เขียว)
 - สถานะ ตรวจแล้ว / ยังไม่ตรวจ / ส่งกลับ
 - กราฟสถิติหน่วยงาน

2. จัดการรายชื่อทั้งหมด
 - ตารางบุคลากรทุกหน่วยงาน
 - ค้นหา / ดูรูป / ตรวจสอบเอกสารแนบ
 - ปุ่ม “อนุมัติ” , “ส่งกลับ” , “พิมพ์บัตร”
3. พิมพ์บัตร / เหมเพลตบัตร
 - เลือกหน่วยงาน / ประเภทบัตร
 - Preview บัตร + ปุ่ม “พิมพ์ PDF”
 - ตั้งค่าตำแหน่ง / ขนาด / สี (เฉพาะในขอบเขตอนุญาต)
4. รายงาน / สถิติ
 - รายงานจำนวนบัตรที่ออกแล้ว แยกตามหน่วยงาน
 - รายงานสถานะการอนุมัติ
 - Export Excel / PDF

บทบาท : **Admin** เหมือน Officer ทั้งหมด แต่เพิ่มเมนู System Setting

เมนูหลักเพิ่มจาก Officer

5. System Setting

- จัดการผู้ใช้ (User / Group / Password Reset)
- ตั้งค่าระบบ (สีบัตร, Logo, QR Key, Path บันทึกไฟล์)
- ตั้งค่าการพิมพ์ / พอร์มัต รายงาน
- สำรอง / คืนค่า ข้อมูล

โครงสร้างการจัดวางแต่ละหน้า

| Topbar: โลโก้ | NPU CeremonyBadge | user |

| Sidebar (เมนู) | เนื้อหา (Dashboard/Form/Table)

| Footer: version / contact info

โครงสร้างข้อมูลหลัก (Core Entities)

หมวด	Entity	หน้าที่ / ข้อมูลหลัก	ความสัมพันธ์
ผู้ใช้และสิทธิ์	users	เก็บบัญชีผู้ใช้ (Submitter, Officer, Admin)	เชื่อมกับทุก entity ผ่าน created_by, updated_by
	departments	หน่วยงานต้นสังกัดของ Submitter	1 Department → หลาย Users
ข้อมูลบุคลากร	staff_profile	ข้อมูลรายบุคคลจากแบบฟอร์ม (ชื่อ-นามสกุล, ตำแหน่ง, หน่วยงาน, หมายเลขบัตร, ประเภทบัตร, สถานะ)	1 Department → หลาย Staff
	photo	รูปภาพที่อัปโหลด (path, metadata, crop x/y/w/h, หมุน deg)	1 Staff → 1 Photo
	badge_request	คำขอออกบัตร (วันที่ส่ง, ผู้ส่ง, สถานะ: DRAFT/SUBMITTED/APPROVED/REJECTED) Request	1 Staff → 1 Badge
บัตรและเทมเพลต	badge	รายละเอียดบัตรจริง (UUID, หมายเลขบัตร, วันที่พิมพ์, ประเภทสี)	1 Badge → 1 Staff
	badge_type	กำหนดสีและคุณสมบัติของบัตร (ชมพู/แดง/เหลือง/เขียว, มีรูปไหม, พื้นสี HEX)	1 Type → หลาย Badge
	badge_template	แม่แบบสำหรับพิมพ์บัตร (ไฟล์เทมเพลต, layout JSON, ตำแหน่ง QR / รูป / ข้อความ)	1 Type → 1 Template
การอนุมัติและบันทึก	approval_log	บันทึก Officer หรือ Admin ที่ตรวจ (เวลา, สถานะ, หมายเหตุ)	1 Request → หลาย Log
	print_log	ประวัติการพิมพ์ (เวลา, ผู้พิมพ์, จำนวนครั้ง, เครื่องพิมพ์)	1 Badge → หลาย PrintLog
รายงาน/ระบบ	report_cache	เก็บผลรวมชั่วคราวสำหรับ Dashboard (เช่น จำนวน บัตรแต่ละสี)	ใช้สำหรับ Dashboard real-time
	system_setting	การตั้งค่าระบบ (โลโก้, คีย์ QR, path ไฟล์, สีธีม)	แก้ไขได้เฉพาะ Admin

สถานะการไหลของข้อมูล (State Flow)

1. Submitter → กรอกข้อมูล → staff_profile + สร้าง badge_request
2. Officer/Admin → ตรวจสอบ → บันทึก approval_log
3. เมื่ออนุมัติ → สร้าง badge และ QR → อ้างอิง badge_type + badge_template
4. พิมพ์บัตร → เพิ่ม print_log
5. Dashboard และรายงาน ดึงจาก report_cache

หลักการสำคัญด้านข้อมูล

- ☐ ทุกตารางควรมี created_at, updated_at, created_by, updated_by
- ☐ id หลักทุกตารางใช้ UUID หรือ AUTO_INCREMENT ตามกรณี
- ☐ ใช้ ForeignKey แบบ PROTECT ในตารางหลัก (ไม่ให้ลบข้อมูลที่เชื่อมกันอยู่)
- ☐ เก็บรูปใน MEDIA_ROOT/photos และบันทึกพารามิเตอร์ใน photo.file_path
- ☐ ใช้ badge_request.status เป็นตัวควบคุม workflow ระหว่าง Submitter ↔ Officer ↔ Admin

ภาพรวม Workflow หลักของระบบ NPU-CeremonyBadge

ลำดับ	ขั้นตอน	ผู้ดำเนินการ	ผลลัพธ์หลัก	สถานะในระบบ
1	ลงทะเบียน / กรอกข้อมูลบุคลากร	Submitter	บันทึกข้อมูลบุคคลใน staff_profile	DRAFT
2	อัปโหลดและปรับรูปภาพ	Submitter	อัปโหลดรูป → crop / หมุน / ตรวจสอบความละเอียด	PHOTO_UPLOADED
3	ตรวจสอบก่อนส่ง	Submitter	Preview บัตรตาม template สีบัตร	READY_TO_SUBMIT
4	ส่งข้อมูลเพื่ออนุมัติ	Submitter	สร้าง badge_request 1 รายการ / ล็อกฟอร์มแก้ไข	SUBMITTED
5	ตรวจสอบข้อมูล	Officer	ตรวจสอบความครบถ้วน / ความถูกต้อง / รูปภาพ	UNDER_REVIEW

ลำดับ	ขั้นตอน	ผู้ดำเนินการ	ผลลัพธ์หลัก	สถานะในระบบ
6	อนุมัติ / ส่งกลับแก้ไข	Officer / Admin	บันทึกผลใน approval_log	APPROVED หรือ REJECTED
7	สร้างบัตร (Badge Generation)	Officer / Admin	สร้าง badge จริง + QR Code (HMAC)	BADGE_CREATED
8	พิมพ์บัตร	Officer / Admin	แปลง PDF จาก badge_template + บันทึก print_log	PRINTED
9	ตรวจสอบรายงาน / Export ข้อมูล	Officer / Admin	รายงานรวมหน่วยงาน / ประเภอบัตร / สถานะ	COMPLETED

Workflow ย่อย (Sub-Flow)

1. Flow Submitter

กรอกข้อมูล → อัปโหลดรูป → Preview บัตร → Submit



(Officer ตรวจ) → อนุมัติ / ส่งกลับ

☐ ทุกครั้งที่ส่งกลับ (REJECTED) → Submitter แก้ไขและส่งใหม่ได้

☐ เมื่อได้รับ APPROVED → ไม่สามารถแก้ไขข้อมูลได้อีก

2. Flow Officer / Admin

รับคำขอ (SUBMITTED)



ตรวจสอบข้อมูล + รูปภาพ



[อนุมัติ] → สร้างบัตร + QR → พิมพ์

[ไม่อนุมัติ] → ส่งกลับ + บันทึกหมายเหตุ

☐ มีหน้ารายการ Queue คำขอ (filter ตามสถานะ / หน่วยงาน / สีบัตร)

☐ หลังพิมพ์ จะเพิ่ม record ใน print_log ทุกครั้ง

3. Flow การพิมพ์บัตร

Badge Template (สีพื้น/ตำแหน่ง)

+

ข้อมูลบุคคล + รูป + QR



PDF (WeasyPrint / ReportLab)



บันทึกประวัติการพิมพ์

☐ พิมพ์ PVC หรือ A4 ได้

☐ มี Preview ก่อนพิมพ์

☐ ระบบตรวจจับ Duplicate พิมพ์ซ้ำ (แจ้งเตือน)

4. Flow รายงาน / Dashboard

ฐานข้อมูล Badge + PrintLog



Aggregate / Cache → Report / Dashboard

☐ ใช้ Chart.js ทำ กราฟสรุป

☐ Officer / Admin ดูข้อมูลภาพรวมได้

☐ Submitter เห็นเฉพาะหน่วยงานตัวเอง

สรุปความสัมพันธ์ของ Workflow กับ ฐานข้อมูล

ขั้นตอน	ตารางหลักที่เกี่ยวข้อง	กิจกรรมสำคัญ
①-④	staff_profile, photo, badge_request	กรอก-อัปโหลด-ส่งคำขอ
⑤-⑥	approval_log, badge_request	ตรวจ-อนุมัติ-ส่งกลับ
⑦	badge, badge_type, badge_template	สร้างบัตร-QR
⑧	print_log	พิมพ์บัตร-บันทึกประวัติ
⑨	report_cache, system_setting	รายงาน-สรุปผล