**🔷 ภาพรวมโครงการ TPRSYSTEM**

**🎯 TPRSYSTEM คืออะไร**

**TPRSYSTEM คือระบบ Web Application ที่คุณพัฒนาขึ้นเพื่อใช้เป็นแพลตฟอร์มบริหารจัดการภายในองค์กรของบริษัท Tripira Co., Ltd. (TPR Group) ซึ่งครอบคลุมงานด้าน:**

* **สถาปัตยกรรม**
* **วิศวกรรม**
* **ภูมิสถาปัตย์ (Landscape)**
* **งานสำรวจเพื่อการออกแบบ**
* **งานประมาณราคาค่าก่อสร้าง**

**โดยมีวัตถุประสงค์เพื่อทำให้การจัดการ งานบุคลากร, โปรเจกต์, ลูกค้า, และ การทำงานภายในองค์กร มีความเป็นดิจิทัลมากขึ้น (Digital Transformation)**

**🧩 ฟีเจอร์หลักของระบบ**

**ระบบแบ่งเป็นโมดูลหลักดังนี้:**

| **โมดูล** | **รายละเอียด** |
| --- | --- |
| **1. Project & Client Management** | **จัดการลูกค้า โปรเจกต์ Timeline ทีมงานเอกสาร ฯลฯ** |
| **2. HRM (Human Resource Management)** | **ระบบพนักงาน, แผนก, วันลา, สิทธิการลา, การอนุมัติ** |
| **3. Attendance System** | **ระบบลงเวลาเข้า-ออกงาน ลา ทำงานนอกสถานที่** |
| **4. Payroll Management** | **ระบบเงินเดือน คำนวณภาษี ประกันสังคม โอนจ่าย** |
| **5. Work Management (Kanban)** | **ระบบจัดการงานรูปแบบ Kanban ให้แต่ละทีม** |
| **6. Client Portal** | **สำหรับลูกค้าเข้ามาดูสถานะโปรเจกต์เฉพาะของตนเอง** |
| **7. Role-based Access** | **จำกัดสิทธิ์เข้าถึงระบบตามบทบาทผู้ใช้ (พนักงาน, HR, หัวหน้า, ลูกค้า, ผู้บริหาร ฯลฯ)** |

**🔧 เทคโนโลยีที่ใช้**

| **ด้าน** | **รายละเอียด** |
| --- | --- |
| **Frontend** | **HTML5, Bootstrap 5, Tailwind CSS (บางส่วน), JavaScript** |
| **Backend/Auth** | **Supabase (Auth + Database + Storage), Firebase Hosting** |
| **Database** | **Supabase PostgreSQL** |
| **Deployment** | **Firebase CLI, Firebase Hosting** |
| **เครื่องมือช่วยพัฒนา** | **Visual Studio Code, ChatGPT, Excel, PowerPoint ฯลฯ** |

**🕳️ สิ่งที่ยังไม่ได้ทำ (To-Do List)**

| **ลำดับ** | **งาน** | **สถานะ** |
| --- | --- | --- |
| **1** | **เชื่อม Dashboard กับ Supabase (แสดงข้อมูลจริง เช่น KPI)** |  |
| **2** | **ปรับระบบ Login ให้รองรับ Verify Email และจัดการ Redirect หลังยืนยัน** |  |
| **3** | **เชื่อมระบบ Attendance กับฐานข้อมูลจริง + GPS / QR check-in** |  |
| **4** | **ระบบอนุมัติวันลา (หัวหน้า - HR) เชื่อม RLS + ฟังก์ชันอนุมัติจริง** |  |
| **5** | **ระบบ Payroll เชื่อมข้อมูลพนักงาน + ลา + เงินเดือน** |  |
| **6** | **ระบบแสดงข้อมูล Client Portal โดยใช้ RLS** |  |
| **7** | **สร้างระบบจัดการสิทธิ์ (RBAC Hybrid) ผ่าน UI** |  |
| **8** | **เชื่อมหน้า departments.html กับ Supabase** |  |
| **9** | **อัปเดตระบบ Notification เช่น แจ้งเตือนลา / เช็คอิน** |  |
| **10** | **ทำระบบอัปโหลดไฟล์งาน (เชื่อม Google Drive API)** |  |

**🧭 แผนดำเนินการถัดไปที่แนะนำ**

1. **✅ แก้ไข Redirect หลังยืนยัน email และ deploy ใหม่**
2. **✅ สร้างหน้า Dashboard ที่เชื่อมข้อมูลจริง (จาก Supabase)**
3. **🛠️ เริ่มเขียนระบบ Role + RLS Rule + Mapping ตำแหน่ง**
4. **🛠️ พัฒนา Attendance & Leave (สำคัญที่สุดใน HR)**
5. **🧪 เริ่มทดสอบ Client Portal จริง (ใช้ RLS จาก auth.uid)**

**✅ สิทธิ์ของระบบ (Access Rights List)**

**🧑‍💼 1. พนักงานทั่วไป (Employee)**

* เข้าดู / แก้ไขโปรไฟล์ส่วนตัว
* ลงเวลาเข้างาน/ออกงาน (Attendance)
* ยื่นคำขอลา (Leave Request)
* ดูประวัติการลาของตนเอง
* ดูสลิปเงินเดือนของตนเอง
* แก้ไขรหัสผ่าน หรือข้อมูลบัญชีตนเอง

**👨‍💼 2. หัวหน้าฝ่าย (Team Lead / Head)**

* ทุกสิทธิ์ของพนักงานทั่วไป +
* ดูรายชื่อพนักงานในทีม/แผนกของตนเอง
* อนุมัติคำขอลาของทีมตนเอง
* ดูประวัติการลา/เข้าออกของทีมตนเอง
* ดูรายงานสรุปการทำงานของทีม

**🧑‍💻 3. ฝ่ายบุคคล (HR Officer)**

* ทุกสิทธิ์ของหัวหน้าฝ่าย +
* เข้าถึงพนักงานทั้งองค์กร
* เพิ่ม/แก้ไข/ลบ ข้อมูลพนักงาน
* ตรวจสอบประวัติการลา / การลงเวลา
* จัดการสิทธิ์การลา / จำนวนวันลา
* จัดการสลิปเงินเดือน
* นำเข้าข้อมูลพนักงานจาก Excel

**🧑‍💼 4. ผู้ดูแลระบบ (Admin / ผู้บริหารระดับสูง)**

* ทุกสิทธิ์ของ HR +
* จัดการสิทธิ์ของผู้ใช้งานทั้งหมด
* เปิด/ปิด RLS Policy
* จัดการข้อมูลองค์กร (เช่น ชื่อบริษัท, โลโก้, หน่วยงาน)
* ดู Dashboard สรุปทั้งระบบ
* ดูรายงานการเงิน เงินเดือน งบประมาณ
* อนุมัติการเปลี่ยนแปลงสำคัญ เช่น ปิดรอบบัญชี, ปิดงวดจ่าย

**🧑‍💼 5. ลูกค้า (Client Portal Access)**

* เข้าดูโปรเจกต์ของตนเองเท่านั้น
* ดู Timeline / การเบิกจ่าย / ความคืบหน้าโครงการ
* ดาวน์โหลดเอกสารงาน
* ส่งข้อเสนอแนะ / แจ้งปัญหา

**👨‍🔧 6. Partner / ผู้รับเหมาช่วง (Subcontractor / External)**

* อัปโหลดไฟล์แบบ/ภาพถ่ายหน้างาน (เฉพาะงานที่รับผิดชอบ)
* ลงบันทึกความคืบหน้าของงาน
* ดูตารางงาน / วันส่งมอบ
* สื่อสารกับทีมงาน TPR ผ่านระบบ

**👑 7. System Superadmin (ภายใน Supabase เท่านั้น)**

* ใช้ service\_role key ทำงานเบื้องหลัง
* อัปเดต user metadata
* ทำ Cronjob, Import Script
* จัดการโครงสร้างฐานข้อมูล

**💰 8. ฝ่ายบัญชี (Finance / Accounting Officer)**

**สิทธิ์หลัก:**

* ดูข้อมูลสลิปเงินเดือน (ของทุกพนักงาน)
* เข้าถึงข้อมูลภาษี ประกันสังคม และการจ่ายเงินต่าง ๆ
* จัดการงวดการจ่ายเงินเดือน (Payroll Cycle)
* อัปโหลด/ดาวน์โหลดใบสำคัญทางบัญชี (เอกสารเบิก, ใบเสร็จ, ฯลฯ)
* เข้าถึงรายงานการเงิน เช่น:
  + รายงานค่าใช้จ่ายพนักงาน
  + รายงานสรุปการจ่ายเงินตามโครงการ
  + รายงานการตั้งเบิกงบประมาณ
* จัดการ Payment Status (จ่ายแล้ว / ค้างชำระ)
* ตรวจสอบสถานะภาษี / หัก ณ ที่จ่าย

**สิทธิ์เพิ่มเติมเฉพาะระบบ Payroll:**

* เพิ่ม/แก้ไขข้อมูลเงินเดือน ฐานเงินเดือน โบนัส ค่าล่วงเวลา ฯลฯ
* อนุมัติรายการคำนวณเงินเดือน (ก่อนจ่ายจริง)
* Export เอกสารจ่ายเงินเดือน (Excel / PDF)
* ส่งข้อมูลให้ธนาคาร (ในกรณีใช้ระบบโอนเงินอัตโนมัติ)

⚠️ **ไม่ได้ให้สิทธิ์ผ่าน UI แต่ใช้ผ่าน Backend Tools เท่านั้น**

**📌 หมายเหตุ**

* สิทธิ์ทั้งหมดควรถูกควบคุมด้วย **RLS Policy** + **JWT Claims** + **Role-based UI Control**
* สิทธิ์สามารถแยกย่อยตาม role, department, company\_id, หรือ project\_id ได้

**✅ แนวคิดระบบสิทธิ์ของ TPR Digital Hub**

**1. สิทธิ์หลัก (Main Roles) ที่มีอยู่**

| **หมายเลข** | **สิทธิ์หลัก (Main Role)** | **คำอธิบาย** |
| --- | --- | --- |
| 1 | admin | สิทธิ์เต็มระบบ |
| 2 | hr | เข้าถึงข้อมูลบุคคล ลา ลงเวลา จัดการพนักงาน |
| 3 | finance | เข้าถึงระบบเงินเดือน การเบิกจ่าย |
| 4 | head | หัวหน้าทีม ใช้อนุมัติการลา, ตรวจสอบทีม |
| 5 | employee | สิทธิ์ทั่วไปของพนักงาน |
| 6 | client | สำหรับลูกค้าที่เข้าสู่ระบบผ่าน Client Portal |
| 7 | partner | สำหรับบุคคลภายนอก เช่น Outsource, Subcontractor |

**2. สิทธิ์ย่อย (Sub-Roles) สำหรับแต่ละประเภท**

ใช้ได้กับ hr, finance, head โดยแบ่งออกเป็น:

| **ระดับ** | **ชื่อสิทธิ์ย่อย** | **คำอธิบาย** |
| --- | --- | --- |
| A | manager | บริหารระดับสูง: ดูและจัดการข้อมูลทั้งหมดในหมวดสิทธิ์นั้น |
| B | head | หัวหน้างานเฉพาะทีม/แผนก |
| C | staff | ผู้ปฏิบัติงานทั่วไป |

**3. การจัดสิทธิ์ (Assignment Strategy)**

| **รูปแบบ** | **รายละเอียด** |
| --- | --- |
| 🧠 อัตโนมัติ | ระบบตรวจสอบจากตำแหน่ง (เช่น ตำแหน่ง: ผู้จัดการบัญชี) แล้วกำหนด sub-role เป็น finance.manager โดยอัตโนมัติ |
| ✋ Manual override | แอดมินสามารถ "ติ๊ก/แก้ไข" สิทธิ์ของผู้ใช้งานจากหน้าจอจัดการ (เช่น ให้พนักงานธรรมดาเป็น head.finance) |

**✅ หน้าบ้าน (Frontend UI) ที่ต้องออกแบบ**

**🔹 หน้า: จัดการสิทธิ์ผู้ใช้งาน (Manage User Roles)**

* **ตารางพนักงาน** (แสดง: ชื่อ, อีเมล, ตำแหน่ง, แผนก)
* **Auto-assigned Role**: แสดงสิทธิ์ที่ระบบวิเคราะห์ให้ตามตำแหน่ง
* **Manual override**: Checkbox หรือ Dropdown สำหรับเลือกสิทธิ์เพิ่มเติม (หรือแทนที่)
* **ตัวอย่าง UI (คร่าว ๆ):**

| **ชื่อพนักงาน** | **ตำแหน่ง** | **Auto Role** | **Sub Role** | **เปลี่ยนสิทธิ์ (ติ๊ก)** |
| --- | --- | --- | --- | --- |
| นางสาว A | ผจก.บัญชี | finance | manager | ☑ admin ☐ hr ☐ head |
| นาย B | วิศวกร | employee | - | ☐ finance ☐ hr ☑ head |

หากคุณต้องการ ผมสามารถ:

* เขียนโครงสร้าง JSON สำหรับเก็บ role/sub-role (ใน Supabase)
* เขียน UI HTML mockup
* สร้างไฟล์ script ดึง/อัปเดต role metadata

คุณอยากให้ผมเริ่มที่ส่วนไหนก่อนครับ? เช่น “UI”, “โครงสร้าง Supabase”, หรือ “script อัปเดต metadata”?

**You said:**

ถ้าผมอยากให้เริ่มต้นจากออกแบบระบบจัดการสิทธิ์ก่อนคุณคิดว่าดีไหมครับ ถ้าดีคุณช่วยอธิบาย ขั้นตอนต่อจากนี้หน่อยครับ

**ChatGPT said:**

การเริ่มจาก **ออกแบบระบบจัดการสิทธิ์ (Permission Management System)** ก่อน **ถือว่า “ถูกต้องและดีมาก” ครับ** เพราะระบบสิทธิ์คือรากฐานสำคัญที่กระทบต่อ:

* ความปลอดภัยของข้อมูล
* การเข้าถึงเฉพาะหน้าที่ที่ได้รับมอบหมาย
* ความชัดเจนในการแสดง/ซ่อนข้อมูล UI
* การควบคุม Role-Based UI/Function Logic ทุกส่วนของระบบ

**✅ ขั้นตอนการดำเนินการ: ออกแบบระบบจัดการสิทธิ์**

**📌 เป้าหมายหลัก**

สร้างระบบที่สามารถ “กำหนด, ปรับ, และควบคุม” สิทธิ์ของผู้ใช้งานได้อย่างยืดหยุ่น ผ่านฐานข้อมูล Supabase และ UI ฝั่ง Frontend

**🔧 Step 1: ออกแบบโครงสร้างข้อมูลสิทธิ์ใน Supabase**

**📄 1.1 สร้าง Table: user\_permissions**

sql

CopyEdit

CREATE TABLE user\_permissions (

id uuid PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id uuid REFERENCES auth.users(id) ON DELETE CASCADE,

role text, -- เช่น 'finance', 'hr', 'admin'

sub\_role text, -- เช่น 'manager', 'head', 'staff'

manually\_overridden boolean DEFAULT false,

created\_at timestamptz DEFAULT now()

);

**📄 1.2 หากใช้ metadata (ไม่สร้าง table แยก)**

ใช้ raw\_user\_meta\_data ใน auth.users เก็บ object เช่น:

json

CopyEdit

{

"role": "finance",

"sub\_role": "manager",

"manual\_override": true

}

**ข้อดี:** ใช้ง่าย, ไม่ต้อง join table  
**ข้อเสีย:** ค้นหา/กรองสิทธิ์จำนวนมากยากขึ้น

**🛠️ Step 2: ออกแบบหน้าจอ UI จัดการสิทธิ์ (Frontend)**

* หน้าใหม่: permissions.html หรือหน้าจัดการสิทธิ์ใน dashboard admin
* ดึงข้อมูลพนักงาน (ชื่อ, ตำแหน่ง, แผนก, อีเมล, role/sub-role)
* UI มี:

| **ชื่อ** | **ตำแหน่ง** | **แผนก** | **Role (อัตโนมัติ)** | **Sub-role** | **✅ เปลี่ยนสิทธิ์** |
| --- | --- | --- | --- | --- | --- |
| เกียรติคุณ | กรรมการผู้จัดการ | บริหาร | admin | - | ✓ finance ✓ hr ✓ head |
| เจนจิรา | ฝ่ายบุคคล | HR | hr | manager | ✗ |

* ปุ่ม “บันทึกการเปลี่ยนแปลง” → ส่งค่าไปอัปเดตใน user\_permissions หรือ user\_metadata

**🔁 Step 3: เชื่อมโยงเข้ากับระบบ Login & Auth (RLS / JWT Claims)**

* เมื่อผู้ใช้ Login → อ่านค่าจาก user\_permissions หรือ user\_metadata
* ฝั่ง Frontend: ซ่อน/แสดงปุ่ม/หน้าจอ ตาม role/sub-role
* ฝั่ง Backend: ใช้ RLS ของ Supabase จำกัดสิทธิ์เชิงข้อมูล (เช่น finance เห็นเฉพาะ payroll)

**🧪 Step 4: ทดสอบการใช้งานจริง**

* ทดสอบพนักงานทั่วไป → เห็นเฉพาะหน้าจอของตน
* ทดสอบ HR → เห็นข้อมูลลา
* ทดสอบ admin → เห็นทุกอย่าง

**✅ สรุปสิ่งที่เราจะทำต่อจากนี้:**

| **ลำดับ** | **งานที่ต้องทำ** | **คุณต้องตัดสินใจ** |
| --- | --- | --- |
| 1 | เลือกโครงสร้างสิทธิ์: จะใช้ table หรือ metadata | ✅ ขอให้คุณเลือก |
| 2 | ให้ผมออกแบบ UI permissions.html (mockup HTML หรือ code จริงก็ได้) | ✅ ขอเลือกได้ |
| 3 | เชื่อมต่อ Supabase → ดึง/แก้ไขสิทธิ์ผู้ใช้ | ต่ออัตโนมัติได้ |
| 4 | ออกแบบ RLS เพื่อควบคุมการเข้าถึงจริงตาม role | ขั้นตอนถัดไป |

**🔁 ความสัมพันธ์กับส่วนอื่นของระบบ**

| **ระบบ** | **บทบาทของ Supabase Auth Users** |
| --- | --- |
| **1. Login System** | ตรวจสอบว่า email+password ถูกต้องหรือไม่ |
| **2. JWT Token** | เมื่อ Login สำเร็จ จะได้รับ JWT Token ซึ่งฝังข้อมูล role, user\_metadata, และ id |
| **3. ข้อมูลสิทธิ์ในระบบ** | ระบบจะใช้ metadata หรือ join กับตารางอื่น เพื่อระบุสิทธิ์ของผู้ใช้ |
| **4. Row-Level Security (RLS)** | ใช้ auth.uid() เพื่อจำกัดการเข้าถึงข้อมูลระดับ row |
| **5. การ Reset Password / ยืนยันอีเมล** | ดำเนินการผ่านระบบ Auth นี้ทั้งหมด (Supabase ส่ง Email ให้) |

**🧩 โครงสร้างข้อมูลที่จัดเก็บใน Supabase Auth > Users**

| **ฟิลด์** | **ความหมาย** |
| --- | --- |
| email | อีเมลที่ใช้เข้าสู่ระบบ |
| encrypted\_password | รหัสผ่านที่เข้ารหัสแล้ว |
| raw\_user\_meta\_data | เก็บข้อมูลเพิ่มเติม เช่น ชื่อ-นามสกุล, ตำแหน่ง, role ฯลฯ |
| created\_at, last\_sign\_in\_at | ใช้ดูประวัติการใช้งาน |
| role (ระบบ) | ค่ามาตรฐาน: authenticated, service\_role, supabase\_admin |
| aud | Audience: ปกติจะเป็น authenticated |
| phone, phone\_confirmed\_at | ถ้ามีระบบยืนยันเบอร์โทร |
| confirmed\_at | วันที่ยืนยันอีเมลแล้ว |
| is\_super\_admin | สำหรับผู้ใช้ระดับสูงสุด (ถ้ากำหนด) |

**🧠 สรุปภาพรวมการทำงานของ Auth Users ในระบบคุณ**

1. **ตอนสมัคร / นำเข้าพนักงาน**
   * ผู้ใช้ถูกสร้างขึ้นใน Auth Users
   * มี metadata บอกว่าใครเป็นใคร
2. **ตอน Login**
   * ระบบส่ง email + password ไปตรวจสอบกับ Auth Users
   * ถ้าถูกต้อง → Supabase คืน JWT Token ที่ฝังข้อมูล metadata
3. **ตอนใช้งานระบบ**
   * ทุกครั้งที่ call API / เข้าหน้าจอ → ตรวจ JWT
   * ระบบ Frontend และ RLS ใช้ metadata/uid เพื่อตรวจสอบสิทธิ์
   * ถ้าเป็น “หัวหน้าแผนก” ก็จะเห็นข้อมูลลูกทีม
4. **ตอนปรับสิทธิ์ / ปรับข้อมูล**
   * ระบบสามารถใช้ script (เช่น update-user-metadata.js) ไปอัปเดต metadata ได้อัตโนมัติ

A diagram of a login

AI-generated content may be incorrect.

https://supabase.com/  
  
Project name  
**tpr-digital-hub**

Project ID  
dvwmsclhfkaivhgvjvff  
  
**SUPABASE\_URL**

https://dvwmsclhfkaivhgvjvff.supabase.co

**Anon public:** eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImR2d21zY2xoZmthaXZoZ3ZqdmZmIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTUxMTMyNzgsImV4cCI6MjA3MDY4OTI3OH0.5n0qDS\_k2gVjvxCsNp6ZO\_zvYTRoo5NE4zKWVMcJoq8

**service\_role:**

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImR2d21zY2xoZmthaXZoZ3ZqdmZmIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc1NTExMzI3OCwiZXhwIjoyMDcwNjg5Mjc4fQ.vmmH7wkMGBBqCdAv0TVp9psxZ9QKraVK6rsAzuO\_rrQ

**Legacy JWT secret**  
WJpMbZAXvp8s9eNr7gLFsQn+4Vb9y8Ai2AGRSWFl07/GYaY+hkMDVqRRB1SDZlR2efdwFZN6bVK+8zjU9J1oGg==

authorization code:

-- เปิด RLS (ถ้ายังไม่เปิด)

alter table public.employees\_with\_roles enable row level security;

# ไปที่โฟลเดอร์โปรเจกต์ที่มีไฟล์ supabase/functions/promote-admin/index.ts

supabase functions deploy promote-admin

supabase secrets set SUPABASE\_SERVICE\_ROLE=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImR2d21zY2xoZmthaXZoZ3ZqdmZmIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc1NTExMzI3OCwiZXhwIjoyMDcwNjg5Mjc4fQ.vmmH7wkMGBBqCdAv0TVp9psxZ9QKraVK6rsAzuO\_rrQ