Act as an Expert Full-Stack Developer and Data Scientist.

ฉันกำลังทำโปรเจกต์ชื่อ **"University Social Listening Platform"** เป็นระบบรับฟังและวิเคราะห์ปัญหาภายในมหาวิทยาลัย โดยระบบนี้จะเป็นแอปพลิเคชัน และมีการใช้ AI มาช่วยวิเคราะห์ข้อมูล

Tech Stack ที่กำหนด:

- **Frontend:** Android (Kotlin)
- **Backend / API:** Python (FastAPI)
- **Database:** MySQL
- **AI/Data Models:** > * **Ollama (Local LLM):** ใช้ทำ NLP ประมวลผลภาษาธรรมชาติ, จัดหมวดหมู่ปัญหาอัตโนมัติ, สกัดเวลาที่เกิดเหตุ (Incident time), และตรวจสอบคำหยาบ
    - **Scikit-learn:** ใช้ทำ K-Means Clustering (Demographic Clustering) เพื่อจัดกลุ่มพฤติกรรมผู้ใช้งาน
    - **Computer Vision:** ใช้ทำ Image Processing ดึงพิกัดสถานที่จากภาพถ่าย

ฟีเจอร์หลักของระบบ:

1. ระบบ Login (แบ่ง Role: นิสิต, บุคลากร, บุคคลทั่วไป) มีการเก็บข้อมูล คณะ, ระดับการศึกษา, อายุ, เพศ และใช้ Hashing ปกป้องข้อมูล
2. กระดาน Feed แจ้งปัญหา (ระบุหมวดหมู่, เวลาเกิดเหตุ, พิกัดอาคาร, และแนบรูปได้)
3. ระบบติดตามสถานะปัญหา (มีเจ้าหน้าที่เข้าตอบและอัปเดตสถานะการแก้ไข)
4. Admin Dashboard (แสดงผล Heat Map, Time Series, และระบบคัดกรองผู้ใช้งาน Toxic/Spam)

งานที่คุณต้องทำในคำสั่งนี้ (Please do this step first): ช่วยออกแบบ **Database Schema สำหรับ MySQL** ที่ครอบคลุมฟีเจอร์ทั้งหมดด้านบน โดยขอเป็นโค้ด SQL (CREATE TABLE) ที่พร้อมใช้งาน มีการกำหนด Primary Key, Foreign Key และ Data Type ที่เหมาะสม พร้อมอธิบายโครงสร้างตารางสั้นๆ ให้ด้วย

🎓 **1. หน้า UX สำหรับ "นิสิต มพ."**

ออกแบบมาเพื่อเก็บข้อมูลเชิงประชากรศาสตร์ (Demographic) สำหรับนำไปจัดกลุ่มพฤติกรรม (Clustering) ครับ

- **หน้า Login:** เข้าสู่ระบบผ่านอีเมลของมหาวิทยาลัย (@up.ac.th)
- **ข้อมูลที่ต้องระบุ:** บัญชีนิสิต, รหัสผ่าน, คณะ (เช่น คณะเกษตรศาสตร์และทรัพยากรธรรมชาติ), ระดับการศึกษา (เช่น ปริญญาตรี), เพศ, และอายุ
- **หน้าแจ้งปัญหา (Create Post):** ระบบจะระบุตัวตนผู้โพสต์เป็น "นิสิต มพ." อย่างชัดเจน
- **การกรอกฟอร์ม:** ต้องเลือกหมวดหมู่ปัญหา (เช่น รถเมล์, อาคาร, การเรียน)
- **การระบุเวลาและสถานที่:** ต้องพิมพ์รายละเอียด, ระบุช่วงเวลาที่พบปัญหา (เช่น 06:00-08:00 น.), และเลือกระบุพิกัดตึก/อาคารที่เกิดปัญหา พร้อมแผนที่จำลอง
- **การแนบไฟล์:** มีฟังก์ชันลากและวาง (Drag and drop) เพื่อแนบรูปภาพจุดเกิดเหตุ ขนาดไฟล์สูงสุด 200MB (ไม่บังคับ)
- **หน้า Feed:** จะแสดงโพสต์ปัญหาต่างๆ พร้อมปุ่มกด "เห็นด้วย" (Upvote) เพื่อดันโพสต์ที่มีคนพบเจอเหมือนกัน

💼 **2. หน้า UX สำหรับ "บุคลากร มพ."**

ออกแบบมาให้รวดเร็วและมีฟังก์ชันการสื่อสารภายในองค์กรเพิ่มเติมครับ

- **หน้า Login:** เข้าสู่ระบบผ่านอีเมลมหาวิทยาลัย (@up.ac.th) เช่นเดียวกัน
- **ข้อมูลที่ต้องระบุ:** จะกระชับกว่าของนิสิต โดยกรอกเพียง บัญชีบุคลากร, รหัสผ่าน, และอายุ
- **หน้าแจ้งปัญหา (Create Post):** ตัวตนจะถูกแสดงเป็น "บุคลากร มพ."
- **หน้า Feed พิเศษ:** มีการแบ่งแท็บหน้า Feed เป็น "ฟีดสาธารณะ" และ "ข่าวสารภายใน"
- **สิทธิ์การตั้งค่าโพสต์ (Privacy):** จุดเด่นคือบุคลากรสามารถตั้งค่าความเป็นส่วนตัวของโพสต์ได้ โดยเลือกได้ว่าจะเป็น "โพสต์สาธารณะ" หรือให้เห็น "เฉพาะบุคลากร" (เหมาะสำหรับการแจ้งข่าวสารภายใน เช่น การตัดไฟตึกอาคาร)

👫 **3. หน้า UX สำหรับ "บุคคลทั่วไป"**

ออกแบบมาเพื่อให้คนนอกหรือบุคคลที่ไม่มีอีเมลมหาวิทยาลัยเข้าถึงระบบได้ง่ายครับ

- **หน้า Login:** จะเข้าสู่ระบบด้วย "เบอร์โทรศัพท์" แทนการใช้อีเมล
- **การสร้างบัญชีใหม่:** หากเบอร์โทรศัพท์นั้นเข้าใช้งานครั้งแรก ระบบจะมีหน้าต่างให้ "ตั้งรหัสผ่านใหม่" และ "ยืนยันรหัสผ่านอีกครั้ง" เพื่อใช้ล็อกอินในครั้งถัดไป
- **การระบุตัวตน:** มีช่องให้ระบุความเกี่ยวข้องกับมหาวิทยาลัย (เช่น "ผู้ปกครองนิสิต")

1. ส่วนการรับข้อมูลและการใช้งานฝั่งผู้ใช้ (User Application)

- **เป็นช่องทางรับแจ้งปัญหาโดยตรง:** ระบบจะเป็นแอปพลิเคชันที่ให้นิสิต บุคลากร และบุคคลทั่วไป สามารถรายงานปัญหาที่พบเจอได้ทันที แก้ปัญหาเดิมที่คนมักไปบ่นในพื้นที่ส่วนตัวหรือกลุ่มปิดแล้วเรื่องเงียบหาย
- **การเก็บข้อมูลแบบเจาะจง:** ผู้ใช้งานสามารถสร้างโพสต์โดยระบุหมวดหมู่ปัญหา เวลาที่เกิดเหตุจริง สถานที่เกิดเหตุ (เช่น ตึกเรียน) และสามารถแนบรูปภาพประกอบได้
- **การปกป้องข้อมูลส่วนบุคคล:** ระบบจะมีการใช้เทคนิค Hashing เพื่อเข้ารหัสและปิดบังข้อมูลส่วนตัวของผู้ใช้งาน
- **การติดตามสถานะ:** (สิ่งที่คุณแพลนจะทำเพิ่ม) ระบบจะมีส่วนให้เจ้าหน้าที่เข้ามาตอบ และมีการเพิ่มสถานะการแก้ไขปัญหา

2. ส่วนการจัดการข้อมูลและแสดงผล (Admin Dashboard)

- **แสดงจุดวิกฤตด้วย Heat Map:** มีหน้ากระดานสำหรับผู้ดูแลที่แสดงจุดที่มีปัญหาเกิดขึ้นหนาแน่น
- **ดูแนวโน้มด้วย Time Series:** กราฟวิเคราะห์เวลาที่เกิดเหตุจริง เพื่อให้เห็นว่าปัญหาเกิดขึ้นมากที่สุดในช่วงเวลาไหน
- **ระบบคัดกรองผู้ใช้งาน:** มีการประเมินสถานะความน่าเชื่อถือของผู้ใช้งาน โดยแยกเป็นผู้ใช้งานปกติและผู้ใช้งานที่มีพฤติกรรม Toxic/Spam

3. ส่วนการวิเคราะห์ด้วยเทคโนโลยี AI (AI & Analytics) ใช้ ollama

- **NLP/LLM (ประมวลผลภาษาธรรมชาติ):** ใช้ AI ช่วยอ่านข้อความเพื่อจัดหมวดหมู่ปัญหา สกัดหาเวลาที่เกิดเหตุจากบริบทข้อความ และนำมาช่วยตรวจสอบคำหยาบ
- **Clustering ด้วย K-Means:** จัดกลุ่มพฤติกรรมผู้ใช้งาน เพื่อหา Insight เชิงลึก เช่น นิสิตคณะ ICT มักเจอปัญหาอะไรที่ตึกไหนบ่อยที่สุด
- **Image Processing:** ใช้สำหรับสกัดพิกัดสถานที่จากภาพถ่ายที่ผู้ใช้แนบเข้ามา
- **Predictive Analytics:** ระบบสามารถวิเคราะห์เพื่อคาดการณ์ปัญหาที่อาจเกิดขึ้นในอนาคตได้

สรุปคือ ระบบของคุณไม่ได้เป็นแค่แอปแจ้งซ่อมทั่วไป แต่เป็น **"แพลตฟอร์มที่ใช้ AI มาช่วยวิเคราะห์พฤติกรรมและปัญหาในมหาวิทยาลัยแบบครบวงจร"** ครับ

- **Image Processing:** ใช้ดึงพิกัดสถานที่จากภาพถ่ายที่ผู้ใช้งานแนบเข้ามาในระบบ