To build the **Hospital Bed and Equipment Availability Tracker** using **Next.js (frontend)**, **FastAPI with Poetry (backend)**, **Langchain RAG (AI component)**, and **PostgreSQL (database)**, I will guide you step-by-step, starting with the database schema structure in PostgreSQL, followed by the backend setup with FastAPI, the AI integration using Langchain RAG, and finally, the frontend in Next.js.

Step 1: **Database Schema in PostgreSQL**

The relational schema will cover **Users**, **Doctors**, **Nurses**, **Beds**, **Medical Equipment**, **Patients**, and **AI Recommendations**.

ER Diagram (Logical Structure)

1. **Users**
    - id (PK)
    - username
    - email
    - role (Admin, Doctor, Nurse, Staff)
    - password_hash (encrypted)
2. **Doctors**
    - id (PK, FK to Users)
    - doctor_type (ENUM: Oncologist, Orthopedic Surgeon, Psychiatrist, etc.)
3. **Nurses**
    - id (PK, FK to Users)
    - nurse_type (ENUM: Emergency Room Nurse, Oncology Nurse, etc.)
4. **Patients**
    - id (PK)
    - name
    - condition (text)
    - admitted_at (datetime)
    - discharged_at (datetime, NULL if still admitted)
    - assigned_bed_id (FK to Beds)
    - assigned_equipment_id (FK to MedicalEquipment)
5. **Beds**
    - id (PK)
    - bed_type (ENUM: ICU, General, etc.)
    - status (ENUM: Available, Occupied, Maintenance)
    - room_number (text)

6. **Medical Equipment**

- id (PK)
- equipment_type (ENUM: Ventilator, Oxygen Tank, Dialysis Machine, etc.)
- status (ENUM: Available, In Use, Under Maintenance)
- location (text)
- maintenance_due_at (datetime)

7. **AI Recommendations**

- id (PK)
- recommended_by (FK to Users)
- patient_id (FK to Patients)
- recommended_bed_id (FK to Beds)
- recommended_equipment_id (FK to MedicalEquipment)
- created_at (datetime)