Assignment: Task Tracker with AI Assistance

Objective

Build a simple **Task Tracker Application** using **.NET Core (GraphQL)**, **PostgreSQL**, **React (Adobe React Spectrum + Relay)**, and **Docker**. The assignment should also demonstrate **how you use AI tools** (e.g., Cursor, Copilot, ChatGPT) to accelerate development.

⏱️ **Estimated Duration:** This task is designed to be completed in **2–3 hours**.

Requirements

Backend (ASP.NET Core + GraphQL + PostgreSQL)

- Define a **GraphQL schema** for `Task` with the following fields:
  - `id` (GUID)
  - `title` (string)
  - `description` (string)
  - `status` (`Pending` / `Completed`)

- Implement GraphQL **mutations**:
  - `createTask(title, description)` → returns created task
  - `updateTaskStatus(id, status)` → returns updated task

- Implement GraphQL **query**:
  - `getAllTasks` → returns a list of tasks

- Use **Entity Framework Core + PostgreSQL** to persist tasks.

Frontend (React + Adobe React Spectrum + Relay)

- Build a UI to:
  - **Add a new task** (form → mutation).
  - **Display all tasks** (query → task list/grid).
  - **Toggle task status** between “Pending” and “Completed” (mutation).

- Use **Adobe React Spectrum** for UI components.
- Prefer **Relay GraphQL client** for data fetching.

Dockerization

- Create **Dockerfiles** for backend, frontend, and database.
- Use **Docker Compose** to orchestrate services:
    - Backend (`.NET Core GraphQL API`)
    - Frontend (`React + Relay`)
    - Database (`PostgreSQL`)
- Running `docker-compose up` should start the entire stack.

Submission

Submit your solution as a **Git repository link**, including:

1. **Source Code** (backend & frontend)
2. **Docker Compose file**
3. **README.md** with:

    - Your **approach and thought process**
    - **AI tools and models used** (e.g., Cursor, ChatGPT, Copilot)
    - **Reflections** on effectiveness: what worked, what didn't, and how AI helped you solve problems

---

Notes

- You are encouraged to **use AI tools** for schema generation, resolver stubs, React components, and Docker setup.
- We are evaluating both **technical ability** and **AI-assisted problem-solving**.
- Keep it simple — a working minimal implementation is enough.