Project Overview

# Frontend

Tech Stack: React with TypeScript and Material-UI

Features Implemented:  
- Incident Dashboard with DataGrid for listing incidents.  
- Sidebar displaying past incidents and recommendations.  
- Detailed Incident Details section with scrollable content.  
- Auto-scrolling to selected incident row.  
- Filtering incidents by default state ("New") on initial load.  
- Beautiful and responsive UI design with consistent layout.  
- Numbering for past incidents and recommendations.  
- Headers fixed while scrolling incident lists.  
- Pagination and sorting capabilities.  
- Environment variable support for backend URL configuration.

# Backend

Tech Stack: Spring Boot with PostgreSQL and pgvector

Features Implemented:  
- REST API to fetch incidents, related incidents, and recommendations.  
- Vector search using PostgreSQL `pgvector` extension.  
- AI-based related incident retrieval using Ollama with `nomic-embed-text` model.  
- Batch job to periodically sync incidents into the vector store every 1 hour.  
- Document storage in vector DB using `spring-ai-starter-vector-store-pgvector`.  
- Deduplication logic to avoid storing duplicate incidents.  
- Embedding conversion between IncidentDTO and Document objects.  
- Automatic Ollama model pulling on startup.  
- Docker configuration for PostgreSQL and Ollama containerization.  
- Proper exception handling and logging.

# Approach

1. \*\*Data Storage and Retrieval:\*\*   
 - Incidents are stored in PostgreSQL and periodically indexed in pgvector.  
 - Vector-based similarity search to find related incidents.  
2. \*\*Frontend Interaction:\*\*  
 - Fetches incidents and recommendations from the backend.  
 - Displays incident details with a smooth UI experience.  
3. \*\*AI Integration:\*\*  
 - Ollama used for local embedding generation.  
 - Free and efficient AI-powered incident retrieval solution.  
4. \*\*Deployment:\*\*  
 - Dockerized backend services.  
 - Frontend deployment with environment variables and build steps.  
 - Simple steps for running the project locally with `npm start` and `docker-compose up`.