Technical Interview Task: Seegnal "Pixel-Perfect" Clone

What You'll Build

A fully functional replica of the **Seegnal eHealth** Login and Home pages — connected to a real local database. The app must run locally on your machine (not deployed to a cloud service).

We want to see clean code, attention to detail, and your ability to deliver a professional, production-quality UI.

Step 1: Study the Reference

Log in to our demo site and carefully study the layout, colors, typography, spacing, and behavior.

URL	https://clinical-react.seegnal.com/
Email	yura.zharkovsky@seegnal.com
Password	Abcd1234!

> **Your goal:** Make your local version look and behave exactly like this site.

Step 2: Frontend (React)

Login Page

- Replicate the visual design exactly (background, inputs, buttons, spacing).
- Validate inputs (e.g., email format).
- Show error states (red borders/text) on invalid input.
- Redirect to the Home Page on successful login.

Home Page (Dashboard)

- **Header:** Replicate the top navigation bar and patient info section.
- **Medication List Panel:**
  - Display a list of the patient's current drugs.
  - Allow **adding** a drug via a search/dropdown input.
  - Allow **removing** a drug from the list.
- **Alerts Panel:**

- When a dangerous drug combination is detected, display alert cards:
  - **Red card** → Major interaction
  - **Yellow card** → Moderate interaction
- Match the alert design from the reference site.

Step 3: Backend (Node.js + Database)

Set up a local server using **Node.js/Express** and a database of your choice (**SQLite, MongoDB, or PostgreSQL**).

Database Tables

Table	Purpose
Users	Stores login credentials (password must be hashed)
Drugs	Stores the drug vocabulary list (see below)
Interactions	Stores rules linking two drugs to a severity level

API Endpoints

Endpoint	What It Does
`POST /login`	Validates credentials against the Users table
`GET /drugs`	Returns the full drug list for the search dropdown
`POST /analyze`	Accepts a patient's drug list, checks the Interactions table, and returns any alerts

---

Step 4: Test Data

Populate your database with the following data to demonstrate the system works.

Drug List

- Sitagliptin 50 mg / Metformin 500 mg
- Metformin 850 mg
- Apixaban 5 mg
- Acalabrutinib 100 mg

- Pantoprazole 20 mg
- Amlodipine 5 mg
- Simvastatin 40 mg

Interaction Rules

Drug Combination	Severity	Alert Message
Simvastatin + Amlodipine	**MAJOR** (Red)	Risk of Myopathy
Apixaban + Acalabrutinib	**MAJOR** (Red)	Increased Bleeding Risk
Pantoprazole (alone)	None	No alert

Step 5: What to Submit

A. GitHub Repository

1. **Frontend** — React source code
2. **Backend** — Node.js/Express source code
3. **Database seed** — A script or dump file to set up the database with the test data above
4. **README.md** — Clear instructions to install dependencies and run the app locally

B. Presentation (5–10 minutes)

Prepare a short, clear presentation to walk us through your work. You will present this during the review session.

What to cover:

- **Overview** — What did you build? A quick demo of the working app (Login → Dashboard → Alerts).
- **Architecture** — How is the project structured? Explain your folder layout, how the frontend talks to the backend, and your database design.
- **Key Decisions** — What tools, libraries, or patterns did you choose and why? (e.g., why SQLite over MongoDB, how you handled state management, etc.)
- **Challenges** — What was the hardest part? How did you solve it?
- **What You'd Improve** — If you had more time, what would you add or do differently?

Format: Slides, a live walkthrough, or a screen recording — whatever works best for you. Keep it concise and focused.

What We're Evaluating

- **Visual accuracy** — How closely does your UI match the reference site?
- **Code quality** — Is your code clean, organized, and maintainable?
- **Full-stack execution** — Does the frontend properly communicate with the backend and database?
- **Attention to detail** — Error handling, validation, and edge cases.

Good luck! If you have any questions, reach out to us before you start.