Organization Chart Builder Challenge (2-Hour Sprint)

Instructions:

Build a minimal, interactive organization chart application from scratch. This exercise tests your ability to create dynamic UI components using React, implement backend API logic with FastAPI, efficiently collaborate with AI tools during development, and prioritize tasks.

Requirements

Backend (FastAPI)

1. **Employee Data Endpoint:**
    - **Purpose: Return a small list of employees (3–10 entries).**
    - **Data Fields: Each employee should include:**
        - **`id` (unique identifier)**
        - **`name`**
        - **`title`**
        - **`manager_id` (refers to another employee's `id` or null if no manager)**
    - **Implementation:**
        - **You can use a simple SQLite database or a csv file.**
        - **The endpoint should respond with JSON.**
2. **Update Manager Endpoint:**
    - **Purpose: Update an employee's manager.**
    - **Request: Accept a JSON payload containing an employee's `id` and a new `manager_id`.**
    - **Response: Return a success message (or an error if the update fails).**
3. **Additional Considerations:**
    - **Keep your API endpoints minimal and focused on the core functionality.**
    - **Ensure proper handling of edge cases (e.g., invalid IDs) in a basic way.**

Frontend (React)

1. **Organization Chart Visualization:**
    - **Display: Render an organization chart showing employee "cards." Each card must display the employee's:**
        - **Name**
        - **Title**
    - **Structure: The layout should visually indicate reporting relationships (e.g., through positioning or simple connectors).**
2. **Drag-and-Drop Functionality:**
    - **Behavior: Enable users to change an employee's manager by dragging an employee card and dropping it onto another card (representing the new manager).**
    - **Library: Use a drag-and-drop library (e.g., react-beautiful-dnd or React DnD).**
    - **Integration: When an employee is dropped onto a new manager, the application should trigger the FastAPI update endpoint.**
3. **UI Feedback:**
    - **Provide a basic loading indicator when the API call is in progress.**
    - **Display a simple inline message or console log for success or error responses.**

Time Constraint

- **Total Time: The complete solution (frontend and backend) must be built from scratch and fully functional within 2 hours (please do not spend more than two hours on this, just document what #TODOs you have left and explain your priorities).**

Deliverables

1. **Source Code:**
    - **Host your solution in a GitHub repository. You may include both the frontend and backend in a single repository or separate them (with clear instructions for each).**
2. **README:**
    - **Setup Instructions: How to run the backend and frontend.**

    - **Technical Choices: A brief explanation of design decisions.**
    - **Time Log: Record the approximate time spent on the backend, frontend, integration, and documentation.**
3. **Code Quality:**
    - **Ensure your code is clean, well-commented, and clearly structured.**
4. **Demo:**
    - **Try and provide some form of demo of your work. A very easy setup for running your code is more than enough.**

Evaluation Criteria

- **Backend/API Logic:**
    - **Correctness and simplicity of the FastAPI endpoints.**
    - **Appropriate use of data storage (SQLite or csv schemas) with proper JSON responses.**
- **Frontend Design & Dynamic Components:**
    - **Effective implementation of the organization chart and drag-and-drop functionality.**
    - **Clarity of UI feedback during API interactions.**
- **Time Management:**
    - **Ability to produce a functional, minimal solution within the 2-hour timeframe.**

Submission

- **Provide a link to your GitHub repository containing the full project.**
- **Ensure that both the frontend and backend have clear setup instructions.**
- **Include your time log and a brief explanation of your AI collaboration experience in the README.**

Recommendation: Start by letting Claude/Gemini/GPT scope out the project structure, break it into steps for you, and only then start.