

Proof of Concept (PoC) for Organization Chart Builder

1. Project Writeup

The Organization Chart Builder App is a web-based tool designed to enable users to create, customize, and manage organization charts seamlessly. Users can add and arrange elements such as employee nodes, department clusters, and connectors to visually represent organizational structures. The application offers dynamic functionality, real-time customization options, and an intuitive interface. With built-in features for node styling, hierarchical layout adjustments, and collaborative editing, users can design organization charts tailored to their specific needs. The app also supports exporting charts as images or PDFs, ensuring users can easily share or integrate them into presentations and reports. Whether for corporate use, team planning, or educational purposes, the Organization Chart Builder App provides a robust and user-friendly solution for creating dynamic organization charts.

2. Components of the Project

The project consists of three main components:

Frontend

- **Framework:** React.js
 - **Description:**
 - Provides an interactive user interface for designing and managing organization charts.
 - Implements attractive functionality for adding, rearranging, and resizing chart elements.
 - Enables real-time customization of nodes, including text, colours, and styles.
 - Offers the option to save or export the final chart as an image or PDF.
 - **Libraries Used:**
 - **React Router:** For navigation between pages, such as the homepage, chart editor, and settings.
 - **React Hooks (useState, useContext):** For managing application state and user interactions.
 - **Basic CSS or TailwindCSS:** For styling and layout, ensuring a responsive and polished UI.
 - **React DnD (react-dnd):** For implementing drag-and-drop functionality.

Backend

- **Optional Backend Integration:**
 - **Features:**
 - User authentication for saving and accessing organization charts.
 - Backend storage for saving templates or shared charts.
 - **Technologies:** Node.js with Express.js or Firebase for serverless backend functionality.

Database

- **Optional Database Integration:**
 - For storing user-created charts or templates.
 - Recommended databases: MongoDB, Firebase Realtime Database, or PostgreSQL.

Hosting Platform

- **Platform:** GitHub Pages or Vercel
 - For static site hosting of the React application.

3. Frontend Components

The frontend is structured using React components:

- **Home.jsx:** A welcoming homepage that directs users to create a new organization chart or view saved charts.
- **MainApp.jsx:** Contains the primary application layout and logic.
- **Toolbar.jsx:** Provides options such as Clear, Save, Delete, and Export (Image/PDF).
- **Sidebar.jsx:** Includes controls for adding employee nodes, departments, and adjusting styles.
- **CanvasArea.jsx:** Serves as the main workspace for building the organization chart.
- **NodeEditor.jsx:** Manages the customization options for individual nodes, including text and color.

4. Backend Components

If backend functionality is added, it will include:

- **User Authentication:** Using Firebase Authentication or Auth0.
- **Data Storage:** Saving and retrieving charts or templates via APIs (Node.js or Firebase).
- **Collaborative Features:** Real-time updates using WebSockets or Firebase Realtime Database.

5. Database Components

- **No Database Integration (Frontend-Only Version):** Chart data is managed dynamically in the frontend state.
- **With Database Integration:** Store chart data persistently for future retrieval or sharing.

6. Hosting Platforms

- **Platform:**
 - GitHub Pages or Vercel for static site deployment.
- **Deployment Process:**
 - Build the project using Vite or Create React App.
 - Deploy using the hosting platform's CLI tools or CI/CD pipelines.

7. Flow Diagram of the Project

User Flow:

1. User visits the app.
2. Selects "New Organization Chart" or opens an existing one.
3. Adds nodes (employees or departments) and connects them to define relationships.
4. Customizes nodes and layout (drag, resize, change color, etc.).
5. Saves or exports the chart (as an image or PDF).

