Project Supervaisor AI

# Title

Build an Interactive Flow Chart Application Using React Flow

# Overview

In this assignment, you will develop a web application using React and the [React Flow](https://reactflow.dev/) library. Your application will enable users to create, manipulate, and connect nodes in a flow chart. The key feature of this project is to design and utilize a structured data model (in JSON format) that represents steps, nodes with sub nodes, and custom attachments (including relationships) between these nodes.

Your application should allow:

* Creating flow charts based on your own designed JSON data structure.
* Attaching parent nodes so that their sub nodes are automatically attached.
* Attaching individual sub nodes to nodes in different steps.
* Defining custom relationships when nodes are attached.

# Learning Objectives

### Integration and Library Usage:

* + Install and integrate React Flow into your React project.
  + Leverage third-party libraries to simplify interactive UI development.

### State Management and Dynamic UI:

* + Manage dynamic state for nodes and edges using React state or external libraries like Redux.
  + Implement real-time updates for adding, moving, and deleting nodes.

### Event Handling and Interactivity:

* + Implement drag-and-drop, node selection, and connection creation.
  + Provide clear visual feedback for user actions.

### Component Architecture and Reusability:

* + Design modular, reusable React components that follow best practices.
  + Organize your code for clarity and maintainability.

### Advanced UI/UX Design:

* + Create an intuitive, responsive interface that works well on various devices.
  + Integrate animations and transitions for a smooth user experience.

### Performance Optimization:

* + Optimize rendering using techniques like memoization and lazy loading.

### Testing and Debugging:

* + Write unit and integration tests for interactive features.
  + Utilize debugging tools to troubleshoot state management issues.

### Data Persistence and Deployment:

* + Implement features to save/load flow charts (using local storage or a backend service).
  + Deploy your application using platforms like Vercel or any other hosting service.

### Documentation and Version Control:

* + Maintain thorough documentation and commit messages using Git or another version control system.

# Assignment Requirements

## Project Setup

* + Create a new React project using Create React App, Vite, or Next.js.
  + Install React Flow and any additional libraries required.

## Build the Basic Flow Chart

* + Render a canvas using React Flow.
  + Pre-load the canvas with initial data driven by your own designed JSON data structure.

*(Note: You are not provided with sample JSON. Think through how to structure your data to include steps, nodes, and sub nodes.)*

## Implement Interactive Nodes and Attachments

* + Allow users to add new nodes and edit existing ones.
  + Enable drag-and-drop functionality for nodes.
  + Implement features to delete nodes and update node information.
  + When a parent node is attached, ensure that all its sub nodes are automatically attached.
  + Support attachments between nodes across different steps.
  + Allow users to define custom relationships when attaching nodes.

## Sidebar/Toolbar and Additional Features

* + Create a sidebar or toolbar with options for:
    - Adding new nodes.
    - Clearing the canvas.
    - Selecting different node types or properties.
  + *(Optional)* Implement undo/redo functionality.

## Data Structure Design

* + Design your own JSON data model to represent:
    - **Steps:** Each with a unique identifier and name.
    - **Nodes:** With properties such as id, label, and an optional list of sub nodes.
    - **Attachments:** Defining relationships between nodes (including parent-child relationships and custom relations).
  + Think carefully about how attaching one node should automatically manage its sub nodes.

## Documentation and Testing

* + Document your code with meaningful comments.
  + Write a detailed README that explains:
    - How to set up and run the project.
    - How your JSON data structure maps to the visual flow chart.
    - Any additional features you have implemented.
  + Test all interactive elements thoroughly to ensure robustness.

# Deliverables

### Source Code:

* + Upload your project to a public GitHub repository named **Supervaisor-Task**.
  + Ensure your repository includes all project files and clear commit messages.

### README File:

* + Include detailed setup instructions and a description of your project.
  + Explain your designed JSON structure and how it drives the flow chart.

### Deployed Application:

* + Deploy your application on a hosting platform such as Vercel (or any other platform).
  + Provide the deployed link as part of your submission.

### Live Demo (Optional):

* + In addition to the deployed link, you may provide a live demo link if available.

# Evaluation Criteria

### Functionality:

* + Does the application meet the requirements (e.g., interactive nodes, dynamic attachments, JSON-driven structure)?

### Code Quality:

* + Is the code modular, well-documented, and maintainable?

### User Experience:

* + Is the interface intuitive, responsive, and visually engaging?

### Innovation:

* + Are additional features (e.g., custom relations, undo/redo) implemented effectively?

### Testing and Documentation:

* + Are there thorough tests and clear documentation on how to set up and use the application?

### Instructions for Submission:

1. Copy the text above and paste it into your preferred word processor (Microsoft Word, Google Docs, etc.).
2. Save the document as “ProjectSupervisorAI.doc” or “ProjectSupervisorAI.docx”.
3. Share the document along with your public GitHub repository (named

**Supervaisor-Task**) and the deployed application link.

Good luck, and happy coding!