Step-by-Step Algorithm

Setup and Initial State:

Initialize a new React project.

- Create a state in the App component to hold the list of employees, the search term, and the currently editing employee.
- Create a CSS file for styling.

Fetch Data from LocalStorage:

- In the App component, use the useEffect hook to fetch employee data from localStorage when the component mounts.
- Parse the stored data from JSON format and set it to the state.

Save Data to LocalStorage:

- In the App component, use the useEffect hook to save the employee data to localStorage whenever the state changes.
- Stringify the state data before storing it in localStorage.

Search Functionality:

- Create a function in the App component to handle the search term input.
- Filter the employee list based on the search term (matching name, surname, ID, or department).

CRUD Operations:

Create a function to add or update employee details:

- Check if an employee is being edited.
- If editing, update the existing employee in the state.
- If adding, create a new employee object with a unique ID and add it to the state.
- Create a function to delete an employee by filtering out the employee from the state.
- Create a function to set the employee to be edited.

Component Structure:

- Create a Header component that includes a search bar.
- Create a Sidebar component for navigation links (add employee, view employees, filter employees).

  Create an EmployeeList component to display the list of employees.
- Create an EmployeeForm component to add or update employee details.

Styling:

- Style the components using CSS to ensure a user-friendly interface.
- Add responsive design elements to handle different screen sizes.

Pseudocode

// Setup and Initial State

```
- Initialize React project
- Create App component with state:
  - employees (list)
  - searchTerm (string)
  - editingEmployee (object)

// Fetch Data from LocalStorage
- Use useEffect hook in App component to fetch employee data from localStorage on mount
- Parse stored data from JSON and set to state

// Save Data to LocalStorage
- Use useEffect hook in App component to save employee data to localStorage on state change
- Stringify state data before storing

// Search Functionality
- Create search function in App component to handle search term input
- Filter employee list based on search term (name, surname, ID, department)

// CRUD Operations
- Create function to add/update employee details:
  - Check if editing employee
  - If editing, update existing employee in state
  - If adding, create new employee object with unique ID and add to state
- Create function to delete employee by filtering out employee from state
- Create function to set employee to be edited

// Component Structure
- Create Header component with search bar
- Create Sidebar component for navigation links (add employee, view employees, filter employees)
- Create EmployeeList component to display employee list
- Create EmployeeForm component to add/update employee details

// Styling
- Style components using CSS for user-friendly interface
- Add responsive design elements for different screen sizes
```