

APP.js

We have created two components: `App.js` and `TodoContainer.js`.

The `App.js` component is responsible for rendering the overall structure of the application, which includes the navigation bar, sidebar, and the main content area. It also manages the state for the search term and the visibility of the sidebar.

The `TodoContainer.js` component, on the other hand, is responsible for displaying the list of todo items based on the selected table and search term. It fetches the data from the API, filters it based on the search term, and passes it down to the child components to be rendered.

By separating the concerns of rendering the overall structure of the application and displaying the todo list, we make the code easier to read and maintain. It also allows us to reuse the `TodoContainer.js` component in other parts of the application, if needed.

This is a React functional component that sets up the application's routing and renders different components based on the current route. Here are the main components used:

- `BrowserRouter` is a component provided by the `react-router-dom` library that uses the HTML5 History API to keep the UI in sync with the URL.
- `Routes` is a component provided by `react-router-dom` that maps the URL path to components.
- `Route` is a component that defines a single route and specifies what component should be rendered when the URL path matches.
- `TodoContainer`, `NavMain`, and `SideBar` are components imported from other files that are used to build the application's UI.

The `useState` hook is used to create state variables for `sideBar`, `currentLink`, and `searchTerm`. These variables are used to control the visibility of the side bar, the current link in the side bar, and the search term entered in the navigation bar, respectively.

Finally, the `return` statement includes the different components of the application structured in the required format. The `NavMain` component is displayed at the top of the page, followed by `SideBar` on the left side of the screen. The main content area is determined by the current route and is displayed in `TodoContainer`. The `Routes` component determines which route is currently active based on the URL path and displays the corresponding `TodoContainer`.

