Design Decisions

- The application uses a component-based architecture to ensure modularity. Components are small, reusable, and focused on single responsibilities.
- React's built-in state management is used for simplicity. For more complex scenarios, consider integrating Redux or Context API in the future.
- CSS modules are employed for component-level styling, which helps prevent style conflicts and promotes reusability.
- Basic error handling is implemented in components to improve user experience and debug potential issues.

Features

Add Tasks:

Users can add new tasks via a form, which includes fields for a title and description.

Edit Tasks:

Tasks can be edited by clicking on them and updating the details in the form.

Delete Tasks:

Tasks can be removed by clicking the delete button associated with each task.

Toggle Completion:

Tasks can be marked tasks as completed or incomplete.

View Completed Tasks:

Users can view the count of completed tasks.

User Authentication:

Allow users to have personal task lists.

Users can register to the system by giving name, email, and password.

Users can login to the system by giving email and password.

Users can see their personal task list.