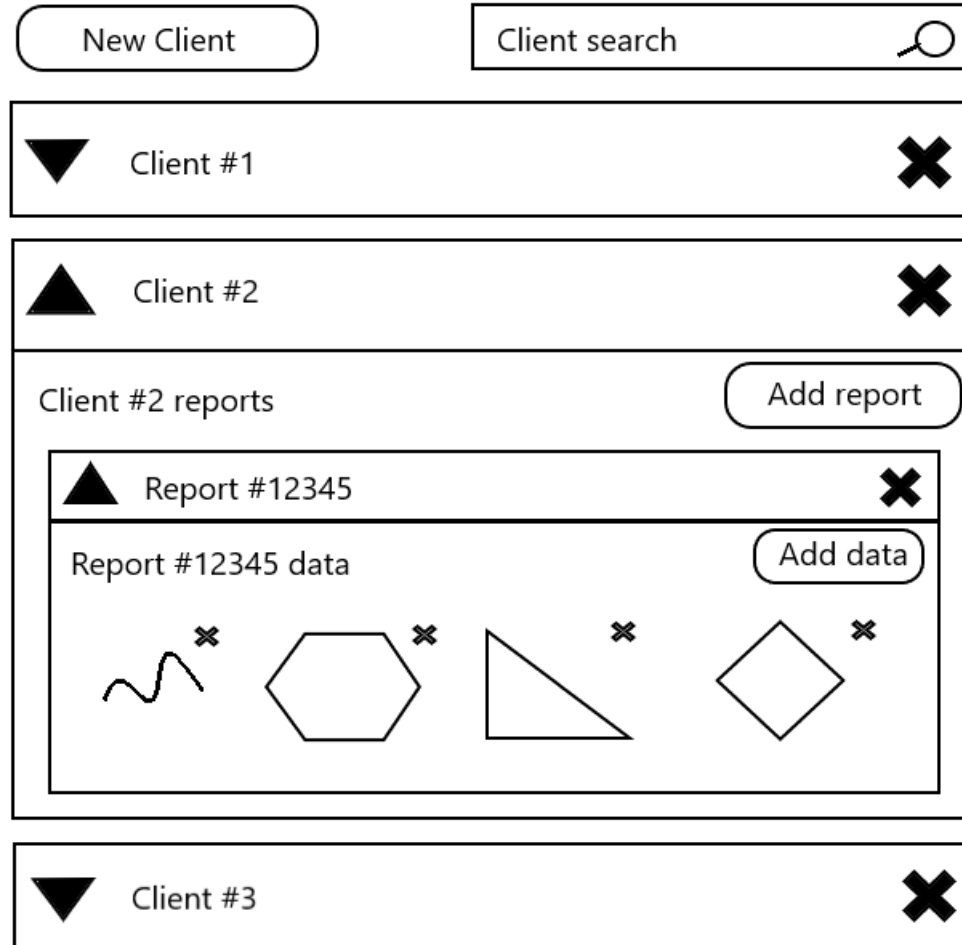


Task overview



You are required to develop a web application using **React, TypeScript, Redux Toolkit, CSS modules, SCSS, and a fake REST API** (e.g., json-server or alternatives) that meets the following requirements:

1. The application should have a list of clients. The client data should be stored separately to emulate receiving it from a web server using a fake REST API.
2. Implement a header with a button that allows adding new clients. The new client's name should be randomly generated. Upon adding a client, a new block representing the client should appear.
3. Each client block should contain an expandable panel with a list of reports.
4. Implement an "Add report" button within each client block. Clicking on this button should add a new report to the corresponding client's list of reports.
5. Each report should have a list of data. Use any type of data visualization (graphs, charts, grids, pictures, etc.) to represent this block.
6. Implement an "Add data" button within each report. Clicking on this button should add a new random piece of data to the corresponding report's list of data.
7. Each client, report, and data item should have a corresponding "x" icon that allows removing it when clicked.
8. Implement a search bar in the header. Clients that do not match the search results should disappear from the list.

9. Utilize Redux Toolkit slices and RTK Query for state management and API communication.
10. Implement automatic data refetching when necessary.
11. Create loading and error state components to handle API calls (e.g., spinners, loading screens, pop-ups).
12. Implement state normalization techniques for efficient data management.
13. Optimize the application by considering edge cases, such as handling scenarios with no data, content overflow, etc.
14. It is acceptable to use third-party libraries and frameworks for CSS (e.g., Bootstrap, Tailwind, Foundation), data visualization components (e.g., Chart.js, ag-grid, Highcharts), and fake data generators.

Please ensure that your code is clean, well-structured, and follows best practices for React, TypeScript, and Redux Toolkit. Focus on creating reusable components and demonstrate your ability to work with data on both the server and client sides.