



Prepare these React.js Low-Level Design (LLD) & you'll be ready for 99% of your interviews.

1. How would you implement infinite scrolling in React?

- Consider the approach for triggering additional data load when the user scrolls near the end of the page or a scrollable element.
- How would you handle throttling, debouncing, or optimization to prevent excessive requests?
- Discuss how to manage state for pagination and loading indicators.

2. How would you implement a search functionality with live filtering in a React application?

- Describe how to optimize the filtering process for large datasets.
- How would you debounce user input to avoid unnecessary re-renders or API calls?
- How would you manage the filtered results state and the interaction with an API?

3. How would you design and implement a form with dynamic fields in React?

- How would you manage form validation, error handling, and state for dynamic inputs?
- Would you use controlled or uncontrolled components for such a form?
- How would you structure the form's state for adding/removing dynamic fields?

4. How would you implement state management for a multi-step form in React?

How would you ensure that data entered in each



Home



My Network



Post



Notifications



Jobs



removing dynamic fields?

4. How would you implement state management for a multi-step form in React?

- How would you ensure that data entered in each step of the form is stored properly and can be accessed across steps?
- How would you manage navigation between steps and handle errors or validation for each step?

5. How would you implement a custom useFetch hook in React for handling HTTP requests?

- Discuss the structure of the hook for handling loading, error, and success states.
- How would you use useEffect to trigger the API request and manage cleanup?

6. How would you implement a draggable list in React?

- Explain how to manage the drag-and-drop state and reordering of list items.
- What would be your approach for handling the drag events, and how would you optimize performance?

7. How would you implement authentication and authorization in React?

- How would you protect routes that require authentication and redirect users to the login page if they're not authenticated?
- How would you implement token-based authentication (e.g., JWT) and handle token expiry?

Check : <https://bit.ly/43sbSk8>

They've helped over 2200+ engineers successfully transition to top product-based companies, even in challenging markets