**Here are top 15 React Machine Coding Interview questions to ace your next frontend interview –**

1. Implement a Todo List: Create a simple todo list application where users can add, edit, and delete tasks.  
  
2. Build a Weather App: Develop a weather application that fetches weather data from an API and displays it to the user.  
  
3. Create a Pagination Component: Design a pagination component that handles navigation between pages of data.  
  
4. Implement a Login Form: Build a login form with validation for email and password fields.  
  
5. Build a Countdown Timer: Develop a countdown timer component that starts from a specified time and decrements until it reaches zero.  
  
6. Create a Modal Component: Design a modal component that can be toggled open and closed.  
  
7. Implement Drag and Drop: Develop a feature that allows users to drag and drop items within a list.  
  
8. Build a Chat Application: Develop a real-time chat application using WebSockets or a library like [Socket.io](http://socket.io/)

9. Create a Dropdown Component: Design a dropdown component with options that can be selected by the user.  
  
10. Implement Form Validation: Develop a form with validation for various fields such as email, password, and phone number.  
  
11. Build a Carousel Component: Design a carousel component that displays a slideshow of images or content.  
  
12. Create a Responsive Layout: Develop a responsive layout that adjusts based on the screen size and orientation.  
  
13. Implement Redux Store: Set up a Redux store and integrate it with a React application to manage state.  
  
14. Build an Infinite Scroll Component: Develop a component that loads more data as the user scrolls down the page.  
  
15. Create a Quiz Application: Develop a quiz application where users can answer multiple-choice questions and see their score at the end.

**1. Weather App**

A weather application that fetches and displays real-time weather information based on the user’s location or input. This project will help you practice API integration, state management, and handling asynchronous data in React.

**2. Todo List**

A classic task management app that allows users to create, read, update, and delete (CRUD) tasks. This project is perfect for mastering React’s state management and component structure, as well as practicing with forms and event handling.

**3. Theme Mode**

A simple yet effective project to toggle between light and dark themes. This will test your understanding of React context and global state management, and how to apply styles dynamically based on user preferences.

**4. Star Rating**

Create a reusable component for a star-based rating system. This project focuses on managing state within a component, handling user interactions, and rendering dynamic UI elements based on the state.

**5. Snake and Ladder Board**

Implement a snake and ladder game board. This complex projectinvolves managing game state, rendering a grid-based board, and implementing game logic, making it an excellent challenge for your problem-solving skills.

**6. Responsive Layout**

Design a layout that adjusts seamlessly to different screen sizes. This project will test your understanding of CSS Flexbox, Grid, and responsive design principles within a React application.

**7. Quiz App**

Create a quiz application where users can create, take quizzes, and receive feedback. This project will help you practice state management, form handling, and conditional rendering in React.

**8. Progress Bar**

Build a dynamic progress bar that visually represents the completion of a task. This project helps you understand how to work with React’s lifecycle methods and how to dynamically update the UI based on state changes.

**9. Pop-Over**

Develop a pop-over component that displays additional information when clicked. This project is great for practicing event handling, conditional rendering, and creating reusable components.

**10. Password Generator**

A secure random password generator. This project will challenge you to implement algorithms for generating random strings and managing state in a simple yet functional UI.

**11. Pagination**

Create a pagination component for navigating through content. This project is essential for understanding how to manage large datasets in React and how to split data into manageable chunks.

**12. OTP Login**

Implement an OTP-based login authentication system. This project will help you understand user authentication, form handling, and secure data transmission in a React application.

**13. Nested Comments**

Create a system for displaying and managing nested comments. This project will challenge your understanding of recursive component rendering and managing complex state structures.

**14. Navbar**

Develop a responsive navigation bar with dropdown menus. This project will test your ability to manage complex layouts, event handling, and conditional rendering based on screen size.

**15. Modal Component**

Build a reusable modal component for displaying overlays. This project is great for practicing component composition, managing state, and handling events in a React application.

**16. Like Button**

Create a like button with animations and state management. This project is perfect for practicing simple state updates and understanding how to incorporate animations into your React components.

**17. Job Board**

Develop a job board application where users can list and search for jobs. This project will help you practice API integration, form handling, and managing complex state within a React application.

**18. Accordion**

Create an accordion component to toggle the visibility of content. This project focuses on managing component state and rendering dynamic content based on user interactions.

**19. Calculator**

Build a basic calculator for arithmetic operations. This project is an excellent way to practice state management, event handling, and implementing logic within React components.

**20. Carousel**

Implement an image carousel for showcasing multiple images. This project will help you understand how to manage component state, handle user interactions, and implement animations.

**21. Countdown Timer**

Create a countdown timer for events. This project will challenge you to manage time-based state updates and implement interval-based state changes in React.

**22. Custom Hook**

Develop a custom React hook for reusable logic. This project will help you understand the power of hooks in React and how to encapsulate reusable logic in a clean, maintainable way.

**23. Drag and Drop Items**

Implement a drag-and-drop feature for items within a list. This project will help you practice advanced user interactions and manage state updates based on complex user actions.

**24. Dropdown Component**

Create a dropdown menu for selecting options. This project focuses on managing component state, handling events, and creating reusable UI components in React.

**25. Form Validation**

Build a form with validation and error messages. This project is crucial for mastering form handling, validation logic, and providing a good user experience through meaningful feedback.

**26. Grid Lights**

Create an interactive grid of lights that can be toggled on and off. This project is great for understanding how to manage complex state and render large grids dynamically in React.

**Questions**

1. Create an app to fetch data from a JSON API. Use this data to render on the page. Provide ability to paginate the results [Video](https://www.youtube.com/watch?v=gnkrDse9QKc&ab_channel=BenAwad)
2. Create a typeahead component.[Video](https://www.youtube.com/watch?v=Kb3YtXDvPo0&lc=UgyEVJO8s1Rs6ukx-up4AaABAg.9RJw-E3qUd79RPuN9QYCQ_&ab_channel=CodingwithKevin)
3. Given a mock, use React to render on the page. Check Sample Mocks section
4. Create a TO-DO app.
5. Infinite scrolling app
6. Create a form that gathers user's data and sends it to server.
7. Create a tic tac toe game.
8. Create a slideshow type app. The data of slides could be given. It contains an image url, and the title of the slide. Left and Right navigation buttons are needed.
9. Create a form app that has a different state of UI when a field is selected, has value or is visited. (Uses onFocus, onBlur events)
10. Create a ticker that increments value every second.
11. Create an app that allows user to click a button 5 times, and times them out afterwards.
12. Create an app that uses debouncing / throttling.