# **100 ChatGPT Prompts that you can use in React.js development**

## **Components and Props**

1. What are the steps to set up a new React.js project using Create React App?
2. How can you define and use a stateful component in React.js?
3. How do you use "props" in React.js to pass data from a parent component to a child component?
4. What is the process to map an array to a list of components in React.js?
5. How do you handle events, such as onClick or onChange, in a React.js component?
6. Describe the steps to use the React.js context API to pass data to nested components.
7. How can you implement conditional rendering in a React.js component?
8. What is the process to make an API call in a React.js component using the useEffect hook?
9. How do you implement form handling and validation in a React.js application?
10. How can you use the React Router library to implement navigation in a React.js application?
11. What is the method to use the useState and useEffect hooks to manage local state in a React.js component?
12. Describe the process to define and use a reducer with the useReducer hook in React.js.
13. How can you implement error boundaries in a React.js application to catch and handle exceptions?
14. What are the steps to use React Portals to render a component into a DOM node outside the current component's hierarchy?
15. How do you use the useMemo hook in React.js to optimize computational intensive functions in a component?
16. What is the method to implement authentication and protected routes in a React.js application?
17. How can you use the useRef hook in React.js to access the properties of a DOM element?
18. How do you use the React Developer Tools extension to debug a React.js application?
19. What is the process to implement internationalization (i18n) in a React.js application?
20. How can you use the styled-components library to style components in a React.js application?

## **State, Rendering, and Inputs**

1. As a frontend developer in [your company], how could you effectively implement and manage state in a class-based component for our [web application]?
2. Given [your company] wants to create a form in our [web application], what could be an approach to manage the form state using controlled components in React.js?
3. Design a detailed process for [your company] to create a stateful function component using the useState hook in our [web application].
4. For [your company's web application], how can you conditionally render a component based on a state variable in React.js?
5. Propose an effective approach for [your company] to handle state changes that depend on the previous state correctly in our [web application] developed with React.js.
6. What could be an efficient method for [your company] to force a component to re-render in our [web application]?
7. As a developer in [your company], how could you differentiate between state and props in React.js when developing our [web application], and when should you use each one?
8. Design a strategy for [your company] to optimize a React.js component's re-rendering performance using the React.memo function in our [web application].
9. Propose a detailed plan for [your company] to synchronize state with local storage in our [web application] developed with React.js.
10. As a frontend developer in [your company], how could you manage a state change for an array or an object in our [web application] developed with React.js?
11. Design a comprehensive approach for [your company] to use the useRef hook in React.js to focus on an input field when a component mounts in our [web application].
12. Given [your company] wants to create a controlled input field in our [web application], what could be the method to do so in React.js?
13. Propose an effective process for [your company] to handle multiple input fields with a single event handler function in our [web application] developed with React.js.
14. Design a detailed plan for [your company] to implement a file upload input field in a React.js component for our [web application].
15. How could [your company] handle radio buttons and checkboxes in a form for our [web application] developed with React.js?
16. Outline a comprehensive strategy for [your company] to validate input fields in a React.js form on submission for our [web application].
17. As a frontend developer in [your company], what could be the method to implement real-time form validation as the user types in our [web application] developed with React.js?
18. Propose a detailed process for [your company] to manage the state of a select box in a React.js form for our [web application].
19. Design a comprehensive approach for [your company] to render a list of input fields and capture their state in our [web application] developed with React.js.
20. Given [your company] wants to manage complex state logic in a form for our [web application], what could be the method to use the useReducer hook in React.js?

## **APIs, Hydration, and localStorage**

1. How can a frontend developer in [your company] effectively interact with a RESTful API using the Fetch API in our [web application]?
2. Propose a method for [your company] to handle API errors gracefully in our [web application], ensuring a good user experience.
3. As a developer in [your company], how would you design an API request abstraction to be used across different components in our [web application]?
4. Outline a comprehensive strategy for [your company] to implement caching of API responses for improving performance in our [web application].
5. Given [your company] needs to interact with a GraphQL API from our [web application], what could be the best practices to do so?
6. Propose a detailed plan for [your company] to integrate a third-party API into our [web application] while ensuring data security.
7. How could [your company] implement optimistic updates in our [web application] to enhance user experience during API interactions?
8. Design a comprehensive approach for [your company] to throttle or debounce API requests from our [web application] to prevent overloading the server.
9. As a frontend developer in [your company], how could you manage API keys securely when interacting with external services from our [web application]?
10. Given [your company] wants to perform server-side rendering (SSR) in our [web application], what could be the steps to hydrate the React.js components correctly?
11. Outline a process for [your company] to implement code splitting with React.js and Webpack in our [web application] to improve initial component hydration time.
12. How can [your company] ensure the preservation of user interaction state during component rehydration in our [web application]?
13. Design a comprehensive strategy for [your company] to handle the hydration mismatch warning in our [web application] developed with React.js.
14. As a developer in [your company], how would you use localStorage in our [web application] to persist user preferences across sessions?
15. Propose a method for [your company] to store and retrieve complex data structures like objects or arrays in localStorage for our [web application].
16. What could be an effective way for [your company] to handle quota exceeded errors in localStorage in our [web application]?
17. Propose a detailed plan for [your company] to implement a fallback mechanism when localStorage is not available in the user's browser in our [web application].
18. How could [your company] use localStorage in our [web application] to implement a simple session-based authentication mechanism?
19. As a frontend developer in [your company], what could be the security considerations when storing sensitive information in localStorage for our [web application]?
20. Design a strategy for [your company] to synchronize state with localStorage in a React.js application, ensuring data consistency in our [web application].

## **APIs, Hydration, and localStorage**

1. As a developer in [your company], how would you implement an error boundary in our [web application] to catch JavaScript errors in a specific part of the component tree?
2. Propose a strategy for [your company] to use error boundaries in our [web application] to gracefully handle component errors without crashing the entire React.js application.
3. What could be an effective way for [your company] to track and report the errors caught by error boundaries in our [web application]?
4. How can [your company] use error boundaries along with higher-order components in our [web application] to encapsulate error handling logic?
5. Design a comprehensive method for [your company] to test error boundaries in our [web application] using Jest and React Testing Library.
6. Outline a step-by-step guide for [your company] to create a spinning loader component in our [web application] that displays while data is being fetched.
7. As a frontend developer in [your company], how would you implement a global loader in our [web application] that is triggered during any API call?
8. How can [your company] use React Suspense to handle the loading state in our [web application], displaying a spinner during data fetch?
9. Propose a method for [your company] to implement a timeout for the spinner in our [web application], switching to a different view if data fetching takes too long.
10. What could be the best practices for [your company] to style the spinner in our [web application] so that it is consistent with our branding?
11. Given [your company] wants to implement a modern and responsive design in our [web application], what could be the effective styling techniques using CSS-in-JS libraries like Styled-components or Emotion?
12. How could [your company] apply theming to our [web application] using Styled-components, allowing for quick switching of visual design?
13. Design a comprehensive strategy for [your company] to structure our CSS styles in the [web application], promoting reusability and maintainability.
14. How can [your company] handle media queries for responsive design in our [web application] using CSS modules?
15. Propose a detailed plan for [your company] to implement CSS animations and transitions in our [web application], enhancing user experience.
16. As a frontend developer in [your company], what could be the considerations and steps to use CSS grid and flexbox for layout design in our [web application]?
17. How could [your company] use Sass or Less preprocessors in our [web application] to improve our CSS workflow?
18. What could be the best practices for [your company] to handle cross-browser compatibility issues in our [web application] related to CSS?
19. As a developer in [your company], how would you manage and update the global styling in our [web application] without causing side-effects?
20. Design a comprehensive approach for [your company] to handle CSS specificity and inheritance in our [web application] to avoid styling conflicts.

## **DNS, Deployment, and Platform as a Service (PaaS)**

1. As a developer at [your company], how would you explain the importance of DNS (Domain Name System) in our React.js web application deployment process?
2. Propose a strategy for [your company] to manage DNS settings when deploying our [web application] to a PaaS like Vercel.
3. What could be an effective method for [your company] to troubleshoot common DNS issues that might arise during the deployment of our [web application]?
4. How can [your company] ensure the secure configuration of DNS for our [web application] when deploying on a platform like Vercel?
5. Can you design a comprehensive guide for [your company] to understand and manage DNS records like A, CNAME, MX, and TXT for our [web application]?
6. Outline a step-by-step plan for [your company] to deploy our React.js [web application] on a PaaS platform like Vercel.
7. As a frontend developer at [your company], how would you set up a continuous deployment pipeline for our [web application] using Vercel and a version control system like Git?
8. How can [your company] utilize the environment variables feature of Vercel for storing and managing sensitive information during the deployment of our [web application]?
9. What could be the best practices for [your company] to handle errors and troubleshoot deployment issues in Vercel for our [web application]?
10. Propose a method for [your company] to implement automatic build and deploy process for our [web application] in a PR-based workflow using Vercel.
11. How could [your company] leverage Vercel's Preview Deployments feature for previewing changes in our [web application] before pushing them to the production environment?
12. Design a comprehensive strategy for [your company] to manage deployment permissions and access controls in Vercel for our [web application].
13. How can [your company] optimize the build time of our [web application] while deploying on a platform like Vercel?
14. As a developer in [your company], what could be the considerations and steps to use a custom domain with SSL for our [web application] in Vercel?
15. How can [your company] use the Vercel CLI to manage and monitor deployments of our [web application]?
16. Propose a detailed plan for [your company] to roll back to a previous deployment of our [web application] in case of any critical issues with the new version on Vercel.
17. What could be the best practices for [your company] to manage and optimize serverless functions in Vercel for our [web application]?
18. As a frontend developer in [your company], how would you use the analytics provided by Vercel to measure and improve the performance of our [web application]?
19. How could [your company] set up a custom caching strategy for our [web application] when deploying on Vercel?
20. Design a comprehensive approach for [your company] to handle scalability and high traffic scenarios for our [web application] on a PaaS platform like Vercel.