Questions from All Weeks – React Compilation

Q.No	Questions
1	What is React?
2	What are the key features of React?
3	Explain what JSX is and why it's used in React?
4	Why do multiple JSX tags needs to be wrapped in a single root element or a Fragment?
5	How does Babel contribute in React development?
6	What is the difference between props and state in React?
7	Explain useState in functional components. How do you update state and why are immutable updates important?
8	What is this.setState in class components and what is a common pitfall when using it?
9	Differentiate between controlled and uncontrolled components in React forms. When would you use each?
10	What is useEffect in React and what are its common use cases?
11	Explain the purpose of the dependency array is useEffect. What happens if its omitted and empty?
12	When would you use useRef in React? Provide an example.
13	Explain what a stale closure is in the context of React's useEffect and how it can be resolved.
14	What is debouncing and how would you implement it in a React component for an input field?
15	What is React reconcilliation and the Diffing algorithm? How do they work together to optimize UI updates?
16	Why is it recommended to use a stable and unique key prop when rendering lists in React and why should you avoid using index as a key if the list items can change order?
17	What are the different phases of a React component's lifecycle?
18	How can you optimize React component re-renders? Name a few strategies.
19	What is the children prop in React?
20	Can you directly modify state in React? Why or why not?
21	What are React developer tools and how are they useful?
22	What are functional components in React?
23	What are class components in React?
24	When would you choose a functional component over a class component in modern React?
25	What is the purpose of render() method in class components?
26	What is the virtual DOM, and why does React use it?
27	What is npm start and npm create vite@latest used in React development?
28	What are props used for?
29	Are props objects? Explain.
30	How does useEffect with cleanup function work and why is it necessary?
31	What is unmounting in the React component lifecycle?
32	How does state trigger re renders in React?
33	Explain the concept of lexical scoping and closure in JS and how they relate to each other.
34	What are React.memo, useCallback and useMemo used for, and how do they help optimize performance?
35	Explain the concept of "lifting state up" in React. When and why would you apply this pattern?
36	What is the purpose of super(props) in a class component's constructor?
37	When would you not use Index as a Key in a react list?
38	How do you pass data from a parent component to a child component in React?
39	How do you pass a function from a parent component to a child component and why would you do this?
40	What is side effect in React?
41	What is the difference between event.propagation() and event.preventDefault()?
42	Explain React.createElement and how its realtes to JSX.
43	What are some alternatives to JSX if you don't want to use it?
44	What Is the primary difference in how class attributes are handled in JSX compared to standard HTML?
45	How does vite differ from create react app in terms of development server and building?
46	What are events in React and how do you handle them?
47	Can a child component directly modify the props it receives from a parent? Why or why not?
48	When would you use useEffect with an empty dependency array versus omitting the dependency array entirely?
49	How do you pass multiple pieces of data from parent to child using props?
50	What does it mean for useState to be Immutable via its setter?
51	Explain the difference between useMemo and useCallback. When would you choose one over the other?
52	Compare React Hook Form and Formik, focusing on their core philosophy, performance implications, and how they handle component re-renders
53	What is React.memo , and how do useCallback and React.memo work together to prevent unnecessary re-renders?
54	Compare BrowserRouter , HashRouter , and MemoryRouter . When is it appropriate to use each?
55	What are the two fundamental "Rules of Hooks"?
56	Describe a real-world scenario where useMemo is crucial for performance, and provide a code example.
57	What are the two primary use cases for the useRef hook in React?
58	What are React.lazy() and suspense, and how do they work together to implement code-splitting?
59	What is the difference between using a standard tag and React Router's Link component for navigation?
60	How does React Hook Form achieve better performance than many other form libraries?

61	What is a custom hook in React, and what is the primary motivation for creating one?
62	What is the purpose of the outlet component in React Router v6?
63	What does the register function in React Hook Form do?
64	What is a Route guard in Reaact and how can one be implemented conceptually?
65	How do you access dynamic URL parameters in a component such as the productid from /product/:productid?
66	When should you avoid using memoization like React.memo or useMemo?
67	Why would you use useRef to store a value instead of a state variable from useState?
68	What is the purpose of Yup in the context of React forms?
69	How does <NavLink> differe from <Link> and what is its primary use case?
70	In which specific scenerios would you recommend a team use React Hook Form over Formik?
71	What are the three main reasons a React component re-renders, and which one often leads to performance issues in deeply nested component trees?
72	What is the formState object in React Hook Form, and what are some of its useful properties?
73	How would you implement schema-based validation using Yup in React Hook Form?
74	Why can't you call a Hook inside a conditional statement? What is the underlying reason for this rule?
75	What is the useNavigate hook used for in React Router?
76	When would you choose to use the useNavigate hook over the component for routing?
77	How can you use the useRef hook to track the previous value of a state variable, and why is this useful?
78	What is the purpose of the control object from the useForm hook in React Hook Form, and when is it necessary?
79	What are the key design principles and rules you must follow when creating a custom hook in React?
80	Explain why a React component might re-render even if its own state and props have not changed.
81	Explain the difference between controlled and uncontrolled components in the context of form management, and how this choice impacts performance
82	How do you create nested routes in React Router v6, and what role does the outlet component play in this architecture?
83	In what specific scenario does useCallback become critical for performance optimization, particularly in relation to child components?
84	How does React Hook Form use the formState object to optimize re-renders, and why is it better than watching the entire form state?
85	Describe how you would implement route-based code-splitting in a React application to improve initial load time.
86	What is the critical difference in server configuration requirements between BrowserRouter and HashRouter , and how does this influence your choice?
87	What is the purpose of wrapping your form submission handler with the handleSubmit function from React Hook Form?
88	Why might React.memo fail to prevent a re-render for a component even if the data passed in a prop appears to be visually the same?
89	When would you use MemoryRouter instead of BrowserRouter or HashRouter ?
90	Besides expensive calculations, what is another crucial use case for useMemo related to props and child component re-renders?
91	What is the purpose of the key prop in React when rendering lists of elements, and why is using an array index as a key often considered an anti-pattern?
92	What is the purpose of the dependency array in hooks like useEffect , useMemo , and useCallback ? What happens if you omit it?
93	How does the design of React Hook Form (using refs) compare to Formik (using state) in the context of imperative versus declarative programming?
94	What is the purpose of the reset function in React Hook Form, and how can it be used?
95	Since React Router doesn't have a built-in "Route Guard," how would you implement one to protect routes requiring user authentication?
96	What is the React Context API and what specific problem does it solve?
97	When should you AVOID using the Context API, and what are the potential performance implications of overusing it?
98	Explain when you would choose the useReducer hook over useState .
99	What are the three main parts of the useReducer hook's implementation?
100	What problems with traditional Redux does Redux Toolkit (RTK) solve?
101	Explain the concept of a "slice" in Redux Toolkit and what the createSlice function does.
102	How do you read data from the Redux store and dispatch actions in a React component using Redux Toolkit?
103	What is Zustand, and what are its key advantages over a library like Redux Toolkit?
104	Can you explain the Atomic Design methodology and its five distinct levels?
105	What is the difference between a Template and a Page in Atomic Design?
106	Compare a "Feature-Based" folder structure with one based on "Atomic Design." When would you choose one over the other?
107	What are path aliases in a React project, and what problem do they solve?
108	What is a "Headless UI" component, and what are the primary tradeoffs of using a headless library versus a traditional one like Material-UI?
109	Using Headless UI's Disclosure component as an example, explain how it simplifies building a complex component like an Accordion.
110	How can you optimize a Context Provider to prevent re-renders when passing an object or function in the value prop?
111	What is the role of the <Provider> component in both the Context API and Redux?
112	In Atomic Design, provide a concrete example of an "Atom," a "Molecule," and an "Organism" for a typical e-commerce site.
113	Describe a scenario where you would choose Zustand over Redux Toolkit for state management.
114	What are the three core principles of Redux?

115	What is the benefit of encapsulating Context consumption logic inside a custom hook?
116	Explain the data flow in an application using Redux Toolkit, from a user clicking a button to the UI updating.
117	Why is a well-defined folder structure important for a front-end project?
118	In a useReducer implementation, what is the role of the action object passed to the dispatch function?
119	Describe how you would set up path aliases in a Vite-based React project.
120	Why is it considered a best practice to create separate contexts for unrelated state, such as Theme and Authentication?
121	How do you create a store in Zustand, and how does a component subscribe to it?
122	In a feature-based architecture, what is the specific purpose of the /shared directory, and how does its content differ from a feature folder like /products ?
123	What is createAsyncThunk in Redux Toolkit, and what problem does it solve in the context of Redux?
124	What are the primary benefits of adopting the Atomic Design methodology?
125	Your notes state that for state management, Context API should be avoided when a dedicated library is a "better fit." What defines a situation where Redux or Zustand would be a better fit than Context API?
126	What is the "separation of concerns" achieved by a headless component?
127	Your notes mention a "hybrid approach" to folder structure, combining feature-based and another pattern. Describe what this hybrid architecture looks like and why it's a powerful pattern for large applications.
128	According to your notes, what makes Zustand performant out-of-the-box compared to Context API?
129	What are the main terminologies used in Redux Toolkit, such as Store, State, Action, Reducer, and Slice?
130	Your notes list several common headless UI components. Name at least four and explain why they are good candidates for a headless approach.
131	In classic Redux, before hooks were common, how would you dispatch an action or get the current state from the store?
132	What is the purpose of the baseUrl property inside a jsconfig.json or tsconfig.json file when configuring path aliases?
133	What are the three main steps for implementing the Context API?
134	Your notes mention three headless UI libraries: Radix UI, Downshift, and Headless UI. Based on the notes, what is a key distinguishing characteristic of each?
135	In a feature-based folder structure, where would you place an Axios instance configuration or a date-formatting utility, and why?
136	In a useReducer reducer function, are you allowed to mutate the state argument directly? Explain why or why not.
137	Contrast the setup process for Redux Toolkit versus Zustand. What key differences in their APIs make Zustand simpler for small projects?
138	What is the tradeoff mentioned in your notes when using headless components regarding "more control" versus "more decisions"?
139	Your Redux Toolkit notes mention that it uses Immer internally. How does this simplify the process of writing reducers compared to classic Redux?
140	Explain the concept of memoization in React. When can memoization be counterproductive, and when is it truly worth using?
141	What are the primary use cases for React DevTools in debugging and performance analysis?
142	Describe lazy loading in the context of web development and React. Provide examples of resources where lazy loading is beneficial.
143	Explain how code splitting works in React applications, and how React.lazy and Suspense facilitate this.
144	What is the purpose of the loading="lazy" attribute in HTML image tags, and how does it contribute to lazy loading?
145	How can CSS be lazy-loaded or optimized to prevent it from being renderblocking?
146	What is the difference between a dynamic import() in JavaScript and React.lazy() ? When would you use each?
147	What is ESLint and what is its primary purpose in a JavaScript/React project? How does it contribute to code quality?
148	Explain the role of the .eslintrc.json file in an ESLint setup.
149	What is Prettier, and how does it differ from ESLint? Why are they often used together in a project?
150	What is Husky, and what role does it play in maintaining code quality in a Git repository?
151	How does lint-staged work, and why is it often used in conjunction with Husky?
152	What is Vite, and what are its main advantages as a build tool compared to older alternatives like Webpack (as used by CRA)?
153	Explain what a monorepo is in software development. What problem does Turborepo solve in the context of monorepos, especially when combined with tools like Vite?
154	How does Turborepo's caching mechanism work to speed up builds in a monorepo?
155	How does Tailwind CSS differ from traditional component-based CSS frameworks like Bootstrap?
156	What is ShadCN UI, and is it a component library in the traditional sense?
157	When would you choose to use Tailwind CSS directly versus using ShadCN UI?
158	Explain the core difference in development workflow between Tailwind CSS and CSS Modules.
159	Describe the purpose of the cn utility function often found in projects using ShadCN UI and Tailwind CSS. What two specific problems does it solve?
160	What does "a11y" stand for, and what is its fundamental goal in web development?
161	Why is using semantic HTML crucial for web accessibility, and what is a common mistake developers make?
162	Explain what ARIA attributes are and provide an example of the three main types: roles, properties, and states.
163	What is a "focus trap" in the context of accessibility, and why is it essential for components like modals or dialogs?
164	Differentiate between aria-label , aria-labelledby , and aria-describedby .
165	Explain the function of the tabIndex HTML attribute, specifically the difference between values 0 , -1 , and positive integers like 1 .
166	What are the key configuration files required to set up Tailwind CSS in a project, and what is the purpose of each?
167	Explain the purpose of postcss and autoprefixer in a modern frontend build process, particularly with a framework like Tailwind CSS.

168	How would you manually implement a basic focus trap in vanilla JavaScript for an accessible modal?
169	How does Tailwind CSS keep its production bundle size small despite having thousands of utility classes?
170	What is the role of the class-variance-authority (CVA) library, and how is it essential for creating variant-based components like those in ShadCN UI?
171	What is Hot Module Replacement (HMR)? Explain why Vite's HMR is significantly faster than the HMR in traditional bundler-based tools like Webpack
172	Beyond build performance, what are some of the key developer experience (DX) and team collaboration benefits of adopting a monorepo architecture?
173	What is the role of the workspaces feature (from npm or Yarn) in a monorepo setup? How does it help manage dependencies?
174	In a Vite configuration file for a monorepo project, what is the purpose of the resolve.alias option? Provide an example of how you would use it
175	How can you programmatically manage focus in a React component, for instance, to focus an input field when the component mounts? Provide a code example using hooks
176	In an .eslintrc.json file, what is the difference between the extends and plugins properties?
177	What is a .prettierrc file? Explain the purpose of common configuration options like singleQuote and trailingComma .
178	When building an accessible modal component in React from scratch, what are the essential ARIA attributes and behaviors you must implement?
179	In the context of an accessible modal, what is the specific significance of the aria-modal="true" attribute?
180	Clarify the relationship between code splitting and lazy loading. Are they the same concept?
181	When using ShadCN UI components in a project created with Create React App (CRA) instead of Next.js, what manual adjustments are typically required for import paths?
182	Describe the command npx husky add .husky/pre-commit "npm run lint-staged" . What does each part of this command do?
183	When setting up Tailwind CSS manually in an older project, what is a common installation issue related to PostCSS, and how can it be resolved?
184	Describe how lazy loading can be applied to non-component resources like images and iframes. What are the native browser mechanisms for this?
185	What are some common strategies for optimizing the initial page load time of a React application?
186	Explain the concept of "stale closures" in React functional components, specifically with useEffect , and how useCallback or dependency arrays help mitigate them.
187	In the context of React rendering, what is "reconciliation"?
188	What are keys in React lists, and why are they important for performance and correct rendering?
189	When should you consider using the useReducer hook instead of useState for state management in a React component?
190	How do memoization techniques like useMemo and useCallback contribute to performance optimization when dealing with the Context API?
191	What is the purpose of "snapshot testing" in React applications, and when is it typically used?
192	What are the key differences between fetch and Axios for making HTTP requests in a React application?
193	How do you cancel an in-flight fetch request within a React component, and why is this important in a useEffect hook?
194	From a frontend developer's perspective, what are the fundamental differences between REST and GraphQL?
195	Where should you store authentication tokens (like JWTs) on the frontend? Discuss the security trade-offs between localStorage and HttpOnly cookies.
196	How would you implement a PrivateRoute component in a React application to protect routes from unauthenticated users?
197	What is a JSON Web Token (JWT) and what are its three main parts?
198	Explain the difference between stateful and stateless authentication.
199	What is CORS, and why is it important for a frontend developer to understand?
200	What is the purpose of Apollo Client, and how does its useQuery hook simplify data fetching in a GraphQL-powered React app?
201	In what scenarios might you prefer using a REST API over GraphQL?
202	What is the purpose of a refresh token in an authentication system?
203	What is the difference between Node.js and Express.js?
204	Explain what a CSRF attack is and how SameSite cookie attributes can help mitigate it
205	From a frontend perspective, what is the role of middleware in an Express.js application?
206	In a TypeScript project, how can you ensure your frontend types accurately reflect the API responses from the backend?
207	What is an ORM like Prisma, and why would a backend team choose to use it instead of writing raw SQL queries?
208	What is the purpose of the schema.prisma file when using the Prisma ORM?
209	You created a useFetch custom hook in your notes. What are the primary benefits of abstracting data-fetching logic into a custom hook like this?
210	What is the role of the <ApolloProvider> component when setting up Apollo Client in a React application?
211	What is the difference between "database migration" and "database seeding"?
212	How does a frontend application typically make a request to a GraphQL API, and how does this differ from REST?
213	When making a POST request using the native fetch API, why is it important to include the Content-Type: 'application/json' header?
214	What is delegated authorization, and how does OAuth facilitate it? Provide a real-world example.
215	What are over-fetching and under-fetching in the context of APIs, and how does GraphQL address these issues?
216	How would you configure CORS in an Express backend to specifically allow requests from your React development server running on http://localhost:5173 ?
217	What is REST (Representational State Transfer)?
218	How does caching differ between REST and GraphQL?
219	What is the purpose of the express.json() middleware in an Express server?

220	How do you define a schema in GraphQL, and what are its main components?
221	What are resolvers in a GraphQL implementation?
222	Explain the concept of a field-level resolver in GraphQL and when you would use one.
223	What is the function call that actually starts an Express server, and what command do you run to start the server file?
224	What is the purpose of Apollo Client's InMemoryCache ?
225	How do you attach an authentication token to every GraphQL request using Apollo Client?
226	Describe the complete frontend flow for handling token expiration and refresh.
227	What is PostgreSQL?
228	What role does Prisma play when working with a PostgreSQL database, and what is its main advantage in a TypeScript project?
229	What does the npx prisma migrate dev command do?
230	What is the purpose of the npx prisma db seed command?
231	How do you install Express.js in a Node.js project?
232	Explain how you can implement role-based authorization in an Express.js backend by chaining middleware.
233	What is the signal property of an AbortController , and what is it used for?
234	How could you automatically cancel a fetch request when a user switches to a different browser tab, and why might this be useful?
235	Compare how fetch and Apollo Client handle caching.
236	Compare how fetch and Apollo Client handle the state management related to a data request.
237	What is the graphiql: true option used for when setting up a GraphQL endpoint with express-graphql ?
238	What is a "Man-in-the-Middle" (MITM) attack, and what is the primary way to prevent it when transmitting authentication tokens?
239	Why is storing sensitive information like authentication tokens in URL parameters a severe security risk?
240	Explain the purpose of the DATABASE_URL environment variable in a project that uses Prisma and PostgreSQL.
241	How do you define a one-to-many relationship between two models, such as User and Product , in a schema.prisma file?
242	What are the common HTTP methods used in a RESTful API, and what CRUD operations do they typically map to?
243	How do you handle request cancellation in Axios, and how does the error handling differ from fetch ?
244	What is unit testing in the context of a React application, and what is its primary goal?
245	Explain the difference between unit testing and integration testing. When would you choose one over the other?
246	You're converting a React testing suite from Jest to Vitest. What are the main code changes you'd expect to make, based on a simple component test?
247	Describe the "Testing Pyramid" concept using the test types mentioned in your notes. How would you allocate your testing budget across unit, integration, and E2E tests?
248	What is End-to-End (E2E) testing, and what critical business value does it provide that unit and integration tests cannot?
249	Compare and contrast Client-Side Rendering (CSR), Server-Side Rendering (SSR), and Static Site Generation (SSG). For each, describe an ideal use case.
250	What is Incremental Static Regeneration (ISR), and how does it solve the core limitation of traditional Static Site Generation (SSG)?
251	In Next.js, explain the difference between getStaticProps and getServerSideProps . When would you absolutely need to use getServerSideProps ?
252	Explain the roles of getStaticPaths and getStaticProps when creating dynamic routes in Next.js, like /blog/[id] . Walk through the process from build to runtime.
253	Your QA team reports a bug that they cannot reproduce, but it's being reported by real users. How can a frontend monitoring tool like Sentry help you debug this "client-only" crash?
254	What are "Breadcrumbs" in the context of error tracking with Sentry, and how can you add custom breadcrumbs to enrich your error reports?
255	What is Docker, and what is the core problem it solves for development teams?
256	From a developer's perspective, what are the key benefits of using a Dockerbased environment over a traditional local development setup?
257	Describe the purpose of a multi-stage Dockerfile in the context of building a production-ready React application.
258	How would you use docker-compose to orchestrate a full-stack application consisting of a React frontend, a Node.js/Express backend, and a MongoDB database?
259	What is CI/CD, and why is it important in a modern development workflow?
260	Explain how you would set up a GitHub Actions workflow to run linting, formatting checks, and tests on every pull request to the main branch.
261	What are the main advantages of using a deployment platform like Vercel or Netlify over deploying a React app manually?
262	How does a platform like Vercel handle deployments for a Next.js or React application? Describe the typical workflow
263	In a Vitest or Jest test file, what is the purpose of the describe and it blocks?
264	In React Testing Library, what is the role of render , screen , and fireEvent ?
265	E2E tests are often described as "flaky." What does this mean, and what are some common causes of flakiness in a Cypress test?
266	What is rendering in React, and why is it a fundamental concept to understand?
267	What are two popular frontend monitoring tools, and what is a key difference between them?
268	You are designing a system with multiple React frontend applications that all need to communicate with the same backend services. How would you architect the deployment and traffic management for this system?
269	In Next.js, what is the difference between fallback: 'blocking' and fallback: true within getStaticPaths ?
270	In a CI/CD pipeline defined in GitHub Actions, why is npm ci recommended over npm install for installing dependencies?
271	What is the fundamental difference between a Docker Image and a Docker Container?
272	How does Sentry's Error Boundary component help you catch and report UI rendering errors in a React application?

273	What is the recommended naming convention for Cypress test files, and what is the reason for this recommendation?
274	Why is it necessary to use a production-like environment or a platform like Vercel to properly test Incremental Static Regeneration (ISR)?
275	What is the role of the cypress.config file in a Cypress project, and where should it be located?
276	What are the primary pros and cons of Server-Side Rendering (SSR)?
277	In a Vitest or Jest test, what is the purpose of a mock function created with vi.fn() ?
278	From an architectural standpoint, why is it beneficial to deploy the frontend and backend of a full-stack application on separate, specialized platforms?
279	What value does the @testing-library/jest-dom package add to your tests?
280	In a docker-compose.yml file, what is the purpose of the volumes directive, and why is it critical for a database service?
281	What does it mean to run Cypress tests in "headless mode," and what is the primary use case for it?
282	Explain the tracesSampleRate configuration in Sentry. How would you set it for a development environment versus a high-traffic production environment?
283	In a docker-compose.yml file, what is the purpose of the depends_on key?
284	React Testing Library promotes using "user-centric" queries. What does this mean, and why is it considered a best practice for writing resilient tests?
285	What are some of the main drawbacks of deploying a web application manually?
286	In the context of Server-Side Rendering (SSR) or Static Site Generation (SSG), what is "hydration"?
287	Why would you choose a platform like Railway or Fly.io for a backend service instead of a platform like Vercel?
288	What is the purpose of the support folder in a standard Cypress project structure?
289	What does the EXPOSE instruction in a Dockerfile do, and does it publish the container's port to the host?
290	Why is debugging a failing end-to-end test generally more difficult than debugging a failing unit test?
291	What is the recommended best practice for making the matchers from @testinglibrary/jest-dom available across all test files in a project?
292	In a CI workflow, what is the purpose of a command like prettier --check . , and how does it help maintain code quality?
293	Briefly define the three core Docker concepts: Dockerfile, Image, and Container.
294	What fundamental problem in React do Error Boundaries solve?
295	From a user experience perspective, what is the single biggest drawback of pure Client-Side Rendering (CSR)?