Sakshi Gawande • 2nd

SWE | Frontend Developer (Next.js, React.js) | Docker, Express, JavaScript, Redux, MongoDB | T...   **Follow**

4d • 🌐

80% of Candidates Get Rejected in the First 10 Minutes.

Here's why:

 - You know HTML, CSS, and JavaScript

 - You're comfortable with React, Vue, and Angular

 - You've used Redux, Tailwind CSS, and GraphQL.

But then the interviewer hits you with the big one:

"Tell me about a complex UI component you built in production and how you optimized its performance."

And suddenly, your resume isn't enough.

Real talk:

Most candidates get filtered out, not because they don't know the tools,

but because they can't explain how they used them in real-world scenarios.

Here are 10 real-time, real-world questions to help you go from:

❌ I studied this on YouTube to ✅ I've solved this in production

1. **How do you design a scalable**, **reusable component library for a large web application**?

→ Talk about design patterns, atomic design, and Storybook for isolated component testing.

2. **How do you ensure responsiveness in your application across different devices**?

→ Use of Flexbox, Grid, CSS media queries, and tools like CSS-in-JS (Styled Components, Emotion).

3. **How do you implement and optimize a UI component like a modal or a dropdown**?

→ Explain event delegation, useRef, and the importance of closing events and ESC key press handling.

4. **How do you approach managing state in complex React apps**?

→ Explain Redux or Recoil, useReducer, Context API, and when you'd use each based on complexity.

5. **What's your approach to error boundaries and handling unexpected errors in React**?

→ Explain React error boundaries, logging with Sentry, and graceful error handling in UI.

6. **How do you handle routing in single-page applications**?

→ Discuss React Router, dynamic routing, nested routes, and lazy loading of routes.

7. **How do you ensure accessibility in your web applications**?

→ Talk about ARIA roles, semantic HTML, keyboard navigation, and ensuring screen reader compatibility.

8. **How do you handle form validation in a React app**?

→ Discuss Yup, React Hook Form, and server-side validation with form error feedback.

9. **How do you optimize an application for high interactivity** (**e.g.**, **complex forms or real-time updates**)?

→ Talk about debouncing, throttling, and WebSockets for real-time updates, and form libraries like Formik or React Hook Form.