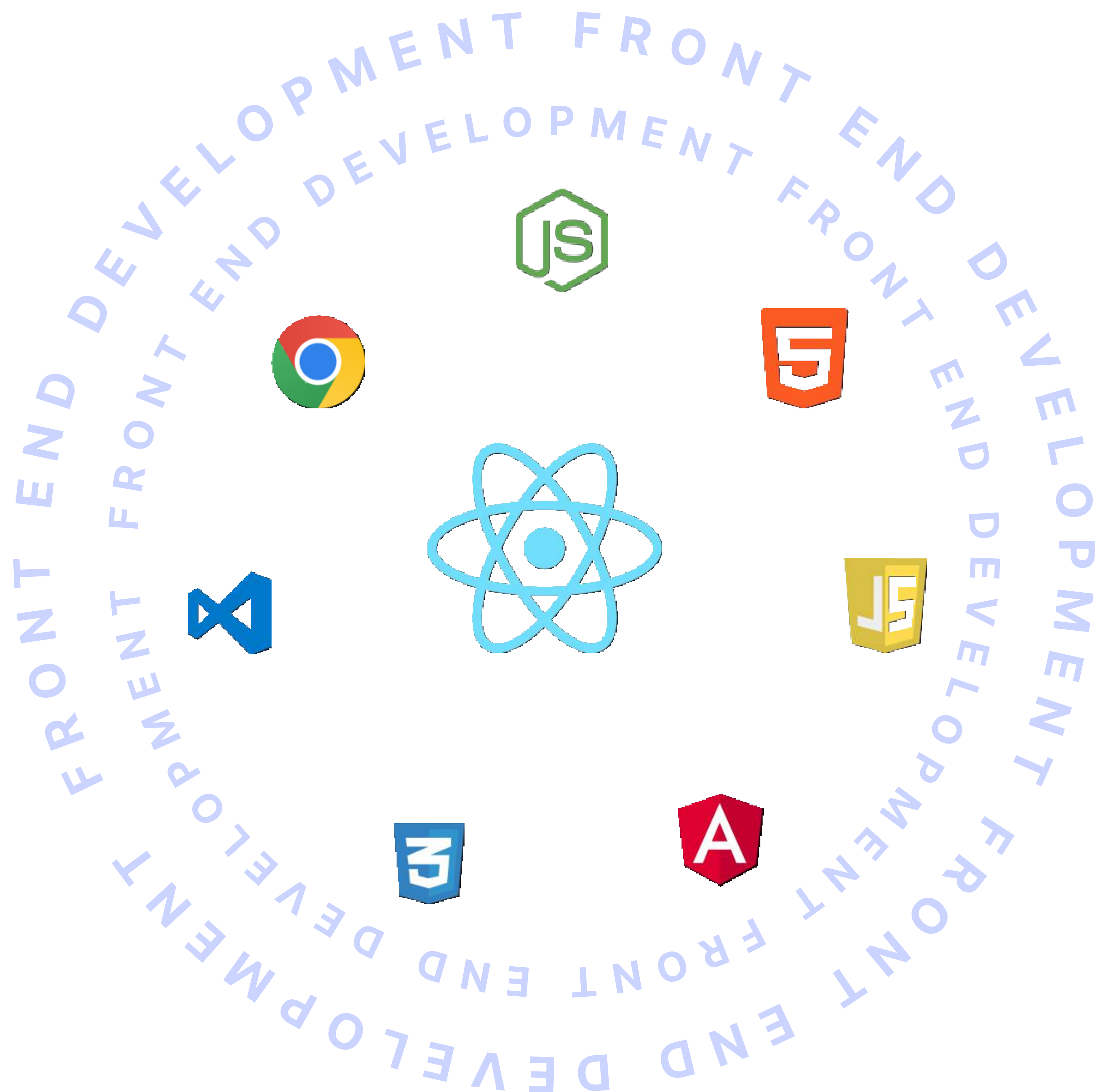
FRONT-END

INTERVIEW QUESTIONS

with Answers to Crack

TECHNICAL INTERVIEW


FRONT END DEVELOPMENT FRONT END DEVELOPMENT FRONT END DEVELOPMENT
FRONT END DEVELOPMENT FRONT END DEVELOPMENT FRONT END DEVELOPMENT

Disclaimer

Prepare for your Front-end Interview with these most asked interview questions.

Make sure to have a thorough knowledge into key front-end technologies, including HTML, CSS, JavaScript, and popular frameworks/ libraries like React, Vue, or Angular.

#1

JavaScript Closures

QUESTION

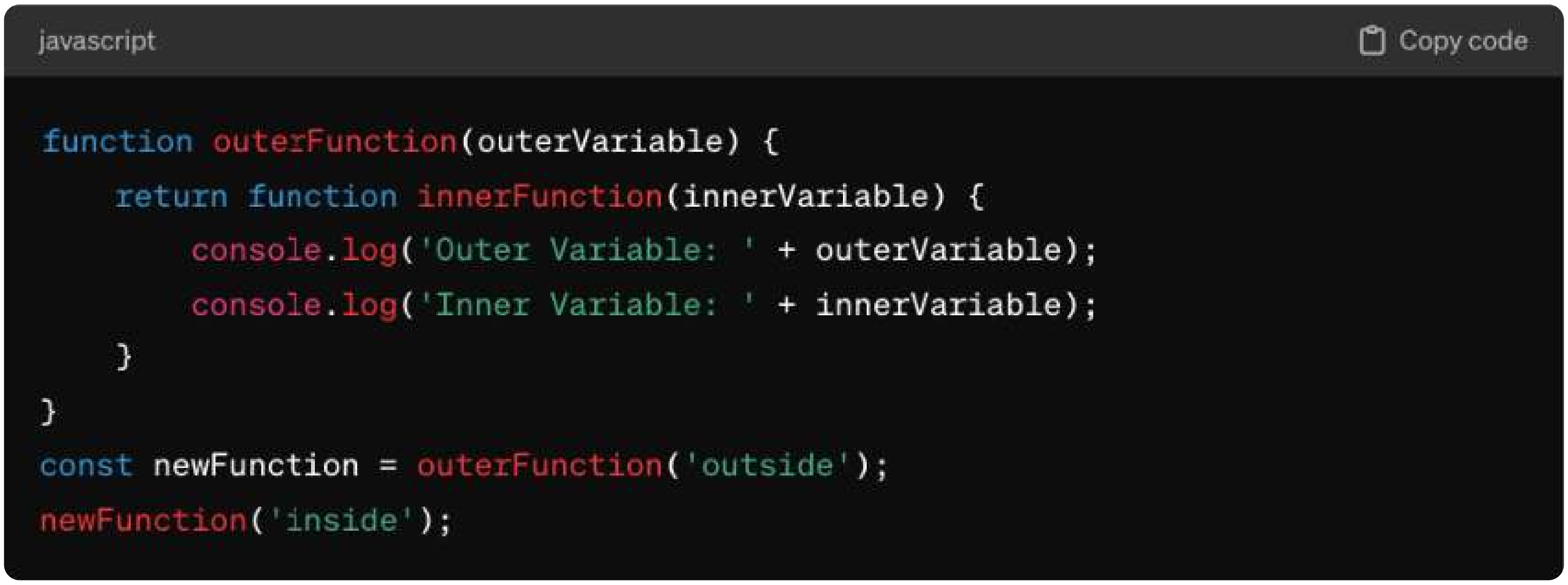
What is a closure in JavaScript? Provide an example.

APPROACH

Explain the concept, then illustrate with a practical example

ANSWER

A closure is a feature where an inner function has access to the outer (enclosing) function’s variables.
Example:



```javascript
function outerFunction(outerVariable) {
    return function innerFunction(innerVariable) {
        console.log('Outer Variable: ' + outerVariable);
        console.log('Inner Variable: ' + innerVariable);
    }
}
const newFunction = outerFunction('outside');
newFunction('inside');
```

#2

Responsive Design

QUESTION

How would you implement a design that needs to be responsive across various devices?

APPROACH

Discuss the use of media queries, flexible grid layouts, and relative units.

ANSWER

Utilize CSS media queries to apply different styles based on device characteristics like width, height, or orientation. Employ a fluid grid layout and use relative units (em, rem, %) instead of pixels to ensure elements scale proportionally.

#3

JavaScript Promises

QUESTION

Explain the difference between `.then()` and `async/await` in handling asynchronous operations in JavaScript.

APPROACH

Compare both, highlighting syntax and usability differences.

ANSWER

`.then()` is used with promises for asynchronous operations, chaining multiple calls for sequential execution. `async/await` makes asynchronous code look synchronous and is syntactic sugar over Promises, improving readability and error handling.

#4

React Component Lifecycle

QUESTION

Describe the lifecycle of a React component and how you would use it to fetch data.

APPROACH

Detail the lifecycle phases and appropriate methods for data fetching.

ANSWER

React components have several lifecycle methods: `constructor(), render(), componentDidMount(), shouldComponentUpdate(), componentDidUpdate(), and componentWillUnmount().` To fetch data, use `componentDidMount()` for class components or `useEffect()` hook in functional components to perform side effects, including data fetching after the initial render.

#5

Cross-Origin Resource Sharing (CORS)

QUESTION

What is CORS and how do you handle it in web applications?

APPROACH

Explain CORS concept and solutions for common issues.

ANSWER

CORS is a security feature that restricts web applications from making requests to a domain different from the one which served the web page. To handle it, configure the server to include CORS headers like `Access-Control-Allow-Origin` in the response, specifying which domains are allowed to access the resources.

#6

Front-End Performance Optimization

QUESTION

What strategies would you employ to optimize the performance of a web application?

APPROACH

List practical techniques for improving front-end performance.

ANSWER

Optimization strategies include minimizing and compressing assets (CSS, JavaScript, images), implementing lazy loading for images and components, using CDN for static assets, optimizing CSS selectors, and leveraging browser caching.

#7

Single Page Application (SPA) SEO Challenges

QUESTION

What are the SEO challenges with SPAs and how can they be addressed?

APPROACH

Identify key SEO issues with SPAs and solutions.

ANSWER

SPAs often struggle with SEO as content is dynamically loaded, making it hard for search engine crawlers to index. Solutions include server-side rendering (SSR), using the History API to update URLs for different views, and leveraging pre-rendering services or static site generators.

#8

Event Delegation in JavaScript

QUESTION

Explain event delegation and its advantages.

APPROACH

Describe the concept and its practical uses.

ANSWER

Event delegation is a technique where instead of adding an event listener to each similar child element, you add a single event listener to a parent element. It leverages the event bubbling phase to catch events from child elements. Advantages include reduced memory usage (fewer event listeners) and dynamically handling events from elements added after the initial page load.

#9

CSS Box Model

QUESTION

Describe the CSS box model and how CSS properties related to it affect layout.

APPROACH

Outline the components of the box model and their impact on element spacing.

ANSWER

The CSS box model consists of four areas: content, padding, border, and margin. These layers affect the total size and spacing of an element on the page. Understanding this model is crucial for precise layout control, as it determines how elements are sized and how they interact with each other in the document flow.

#10

Web Accessibility (a11y)

QUESTION

How do you ensure your web application is accessible?

APPROACH

Discuss key principles and practices for web accessibility.

ANSWER

To ensure web accessibility, follow WCAG guidelines and use semantic HTML elements for structure, ARIA roles for enhanced semantics, ensure keyboard navigability, provide alt text for images, and ensure contrast ratios are sufficient for readability. Testing with screen readers and using accessibility audit tools can help identify and address issues.

#11

React Hooks

QUESTION

Compare the useState and useReducer hooks in React.

APPROACH

Highlight differences and use cases for each.

ANSWER

`useState` is suitable for simple state logic, providing a variable and function to update it. `useReducer` is better for complex state logic that involves multiple sub-values or when the next state depends on the previous one, as it lets you manage local state of complex components with a reducer function.

#12

Tree Shaking in Webpack

QUESTION

What is tree shaking and how does it help in front-end development?

APPROACH

Explain the concept and its benefits.

ANSWER

Tree shaking is a process used in modern build tools like Webpack to eliminate unused code from the final bundle. It improves performance by reducing the size of the JavaScript files that browsers need to load, parse, and execute, leading to faster page load times.

#13

CSS Flexbox vs. Grid

QUESTION

Compare CSS Flexbox and Grid. When would you use one over the other?

APPROACH

Highlight the strengths and use cases of each layout model.

ANSWER

Flexbox is a one-dimensional layout method ideal for arranging items in a single row or column, offering control over alignment and spacing. Grid is a two-dimensional layout system, perfect for creating complex layouts and aligning content within rows and columns. Use Flexbox for simpler, linear layouts and Grid for more complex, multi-dimensional layouts.

#14

Progressive Web Apps (PWAs)

QUESTION

What makes a web application a PWA?

APPROACH

List essential features and benefits of PWAs.

ANSWER

PWAs are web applications that use modern web technologies to provide a fast, reliable, and engaging user experience. Key features include the ability to work offline, receive push notifications, and access hardware features, which can improve engagement and performance on mobile devices.

#15

Web Performance Metrics

QUESTION

What key performance metrics would you consider critical for a web application?

APPROACH

Identify and explain important web performance metrics.

ANSWER

Critical web performance metrics include First Contentful Paint (FCP), Largest Contentful Paint (LCP), Time to Interactive (TTI), Speed Index, Cumulative Layout Shift (CLS), and First Input Delay (FID). These metrics help understand the loading experience, interactivity, and visual stability of a page.

#16

Security in Front-End Applications

QUESTION

How do you protect a front-end application from common security threats?

APPROACH

Discuss strategies and practices to enhance security.

ANSWER

To protect a front-end application, implement Content Security Policy (CSP), sanitize user input to prevent XSS attacks, use HTTPS for secure communication, store sensitive data securely, and keep dependencies up to date to avoid vulnerabilities.

#17

Server-Side Rendering vs. Client-Side Rendering

QUESTION

Compare server-side rendering (SSR) and client-side rendering (CSR).

APPROACH

Outline differences and discuss when to use each.

ANSWER

CSR renders pages directly in the browser using JavaScript, leading to faster subsequent page loads and a smoother user experience. SSR generates HTML on the server and sends it to the client, improving initial load times and SEO. Use CSR for dynamic, app-like experiences and SSR for static sites or when SEO is a priority.

#18

State Management in SPA

QUESTION

How do you manage state in a single-page application (SPA)?

APPROACH

Describe common state management patterns and tools.

ANSWER

In SPAs, state management can be handled with global state management libraries like Redux or Context API in React applications. These tools help maintain consistency across the application, making it easier to share state across components and manage changes reactively.

#19

Front-End Testing Strategies

QUESTION

What strategies and tools do you use for front-end testing?

APPROACH

Discuss different types of testing and tools for front-end applications.

ANSWER

Effective front-end testing strategies include unit testing with Jest, integration testing with React Testing Library, and end-to-end testing with Cypress or Selenium. Each testing type targets different aspects of the application, from individual functions and components to integrated workflows and the full user experience.

#20

Implementing Dark Mode

QUESTION

How would you implement a dark mode feature in a web application?

APPROACH

Outline a technical approach to support dark mode.

ANSWER

To implement dark mode, use CSS custom properties (variables) for color schemes and media queries (prefers-color-scheme) to detect system theme preferences. Provide a toggle option for users to switch manually, storing their preference locally to persist across sessions.

These questions represent a mix of foundational knowledge and practical skills in front-end development, reflecting the breadth and depth expected in interviews at top tech companies. Remember, the key to success lies not only in knowing the answers but in understanding the underlying concepts and being able to discuss and apply them in various scenarios.