**🟢 HTML (5 Questions)**

1. **What is the purpose of the**alt**attribute in**<img>**tags?**
   * Improves accessibility and SEO; displays text if the image fails to load.
2. **How does the**<picture>**tag differ from**<img>**?**
   * <picture> allows multiple sources (e.g., different resolutions/formats) for responsive images.
3. **What are HTML data attributes? How would you use them?**
   * Custom attributes (data-\*) store extra info for JavaScript (e.g., data-user-id="123").
4. **Explain the difference between**<iframe>**,**<embed>**, and**<object>**.**
   * All embed external content, but <iframe> is for web pages, <embed> for plugins, and <object> for general media.
5. **How do you lazy-load images in HTML?**
   * Use loading="lazy" in <img> or <iframe> to defer offscreen content loading.

**🟡 CSS (10 Questions)**

1. **What is BEM (Block-Element-Modifier) in CSS?**
   * A naming convention (e.g., .block\_\_element--modifier) to improve maintainability.
2. **How does**flexbox**differ from**grid**?**
   * Flexbox is 1D (rows/columns); Grid is 2D (rows + columns simultaneously).
3. **What is the**z-index**property? When does it not work?**
   * Controls stacking order. Fails if the element lacks position (relative/absolute/fixed).
4. **Explain CSS**will-change**property.**
   * Hints browsers about future changes (e.g., will-change: transform) for performance optimizations.
5. **How do you implement dark mode in CSS?**
   * Use prefers-color-scheme media query or toggle a class with JavaScript.
6. **What is a CSS preprocessor? Name examples.**
   * Tools like SASS/LESS that add variables, nesting, and mixins to CSS.
7. **How do you center a div vertically and horizontally?**

css

Copy

Download

.center {

display: grid;

place-items: center;

}

1. **What are CSS variables (custom properties)?**
   * Reusable values (e.g., --main-color: #fff;) accessed via var(--main-color).
2. **How do you handle CSS specificity conflicts?**
   * Use !important sparingly, increase specificity, or refactor selectors.
3. **What is the**cascade**in CSS?**
   * Rules for resolving conflicts (source order, specificity, importance).

**🔵 JavaScript (15 Questions)**

1. **What is the event loop in JavaScript?**
   * Mechanism handling asynchronous callbacks via the call stack and task queue.
2. **Explain**map**,**filter**, and**reduce**.**
   * map: Transforms array items.
   * filter: Returns items matching a condition.
   * reduce: Aggregates array into a single value.
3. **What is the rest/spread operator?**
   * ... spreads array/object elements or collects arguments into an array.
4. **How does**async/await**differ from promises?**
   * Syntactic sugar for promises; makes asynchronous code look synchronous.
5. **What is a pure function?**
   * A function with no side effects (same input → same output).
6. **Explain**debounce**vs.**throttle**.**
   * **Debounce**: Delays execution until after a pause (e.g., search input).
   * **Throttle**: Limits execution rate (e.g., scrolling events).
7. **What is the**Proxy**object in JavaScript?**
   * Creates a wrapper to intercept/trap object operations (e.g., get/set).
8. **How do you deep clone an object?**
   * JSON.parse(JSON.stringify(obj)) or libraries like Lodash.
9. **What is the Temporal Dead Zone (TDZ)?**
   * The period where let/const variables exist but are uninitialized.
10. **How does**Object.freeze()**work?**
    * Prevents modifications to an object (no adds/deletes/changes).
11. **What is Web Workers API?**
    * Runs scripts in background threads to avoid blocking the UI.
12. **Explain**WeakMap**and**WeakSet**.**
    * Collections holding weak references (don’t prevent garbage collection).
13. **What is tail call optimization?**
    * Compiler optimization to reuse stack frames for recursive functions.
14. **How do you detect memory leaks in JavaScript?**
    * Chrome DevTools’ Memory tab or performance.memory API.
15. **What is a generator function (**function\***)?**
    * Functions that can pause/resume execution using yield.

**🟠 React (10 Questions)**

1. **What is the difference between**useMemo**and**useCallback**?**
   * useMemo caches computed values; useCallback caches functions.
2. **How does React Router work?**
   * Manages navigation via <BrowserRouter>, <Route>, and <Link>.
3. **What is prop drilling? How do you avoid it?**
   * Passing props through multiple layers. Fix with Context API or state management.
4. **Explain React Fragments (**<> </>**).**
   * A way to group elements without adding extra DOM nodes.
5. **What is React.StrictMode?**
   * Highlights potential issues (e.g., unsafe lifecycle methods) in development.
6. **How do you handle forms in React?**
   * Controlled components (React state) or libraries like Formik.
7. **What is server-side rendering (SSR) in React?**
   * Rendering React on the server for faster initial loads (e.g., Next.js).
8. **What are React Portals?**
   * Render children outside the parent DOM hierarchy (e.g., modals).
9. **How do you test React components?**
   * Jest + React Testing Library or Enzyme.
10. **What is React Fiber?**
    * A reimplementation of React’s core algorithm for better performance.

**🟤 Node.js (5 Questions)**

1. **What is the**EventEmitter**class in Node.js?**
   * Handles custom events via .on() (listen) and .emit() (trigger).
2. **How do you manage environment variables in Node.js?**
   * Use the dotenv package or native process.env.
3. **What is clustering in Node.js?**
   * Spawning multiple processes to utilize CPU cores (via cluster module).
4. **Explain JWT (JSON Web Tokens) authentication in Node.js.**
   * Stateless tokens for auth (created via jsonwebtoken package).
5. **What is the purpose of**helmet**in Express.js?**
   * Middleware to secure HTTP headers (e.g., prevent XSS, clickjacking).

**🟣 MongoDB (5 Questions)**

1. **What is sharding in MongoDB?**
   * Splits data across multiple servers for scalability.
2. **How do you perform transactions in MongoDB?**
   * Use session.startTransaction() with commitTransaction or abortTransaction.
3. **What is the aggregation pipeline?**
   * A framework for data processing (e.g., $match, $group, $sort stages).
4. **How does MongoDB ensure high availability?**
   * Replica sets (primary/secondary nodes) with automatic failover.
5. **What is the difference between**find()**and**findOne()**?**
   * find() returns a cursor (multiple docs); findOne() returns a single document.