Client & Server Side Rendering.txt

2. Explain the key differences and use cases between React's Client-Side Rendering (CSR) and Server-Side Rendering (SSR). Provide examples of scenarios where each approach is advantageous, and discuss the challenges associated with using React in both contexts.

Ans: In React, Client-Side Rendering (CSR) loads a bare HTML page and relies on JavaScript to render the content on the client’s browser, enhancing interactivity. CSR suits applications where user engagement is prioritized, such as single-page applications (SPAs) and dashboards. For example, a social media feed benefits from CSR since only required data is loaded dynamically, improving client-side reactivity.

Server-Side Rendering (SSR), however, generates HTML on the server and sends fully rendered content to the client. This method is advantageous for SEO-sensitive applications, like e-commerce sites, where faster page loads and initial content visibility are crucial for search engine indexing and user experience.

Challenges include managing state across client and server in SSR, which can be complex. For CSR, the initial loading time can be slower, especially with larger JavaScript bundles, and may affect performance on lower-end devices. Balancing SEO, performance, and user experience often requires hybrid approaches, like React’s Next.js.