**1. Scenario: E-Commerce Cart Management**

* *Explanation*: In an e-commerce site, users can add items to a cart, update quantities, and remove items.
* **Question**: How would you manage the state of the cart using React? What state management library would you prefer, and why?

**2. Scenario: Dynamic Form Rendering**

* *Explanation*: A multi-step form requires fields to be displayed dynamically based on user input in previous steps.
* **Question**: How would you implement a dynamic form in React? How would you handle validation and conditional rendering?

**3. Scenario: Handling API Errors**

* *Explanation*: A dashboard fetches data from an API and needs to display a loading spinner, handle errors, and show the data.
* **Question**: How would you implement error handling in React while fetching data from an API? How would you differentiate between different types of errors (e.g., network error vs. server error)?

**4. Scenario: Real-time Chat Application**

* *Explanation*: A real-time chat application requires updating the UI with new messages as they arrive.
* **Question**: How would you implement real-time updates in React? Would you use WebSockets or any other method? Why?

**5. Scenario: User Authentication**

* *Explanation*: After a user logs in, their information should be accessible across the application.
* **Question**: How would you handle user authentication in a React app? How would you protect routes based on user roles?

**6. Scenario: Dark Mode Toggle**

* *Explanation*: A website has a button to toggle between light and dark modes.
* **Question**: How would you implement a theme toggle feature in React, ensuring the user's preference is remembered across sessions?

**7. Scenario: Infinite Scrolling**

* *Explanation*: A social media feed loads more posts as the user scrolls down.
* **Question**: How would you implement infinite scrolling in React? How would you handle performance optimizations for large data sets?

**8. Scenario: Search Autocomplete**

* *Explanation*: A search bar provides suggestions as the user types.
* **Question**: How would you implement a search autocomplete feature in React? How would you optimize it to avoid unnecessary API calls?

**9. Scenario: File Upload with Progress Bar**

* *Explanation*: A user uploads a file, and a progress bar shows the upload status.
* **Question**: How would you implement file uploads in React, showing progress to the user?

**10. Scenario: Localization and Multi-language Support**

* *Explanation*: The application should display content in different languages based on user preference.
* **Question**: How would you implement localization in a React app? How would you handle dynamic language switching?

**11. Scenario: Role-Based Dashboard**

* *Explanation*: An admin dashboard displays different components based on user roles.
* **Question**: How would you design a React component that renders different layouts based on user roles? How would you secure role-based access?

**12. Scenario: Form Auto-Save**

* *Explanation*: A form auto-saves user input periodically to avoid data loss.
* **Question**: How would you implement auto-save functionality in React? How would you ensure efficient use of resources?

**13. Scenario: Drag-and-Drop File Manager**

* *Explanation*: A file manager allows users to drag and drop files into folders.
* **Question**: How would you implement drag-and-drop functionality in React? Which libraries or techniques would you use?

**14. Scenario: Responsive Sidebar Navigation**

* *Explanation*: A website has a sidebar that collapses on smaller screens.
* **Question**: How would you implement a responsive sidebar in React? How would you ensure a smooth user experience during transitions?

**15. Scenario: Collaborative Document Editing**

* *Explanation*: A collaborative tool allows multiple users to edit a document in real-time.
* **Question**: How would you handle real-time data synchronization in React for a collaborative application? What challenges might arise, and how would you address them?