React – Json-Server And Firebase Real Time Database

**Question 1: What do you mean by RESTful web services?**

RESTful web services are APIs that follow REST principles. REST (Representational State Transfer) is an architecture that allows communication between a client (like a browser or mobile app) and a server. These services use standard HTTP methods:

* GET: To retrieve data.
* POST: To send data to the server.
* PUT: To update data on the server.
* DELETE: To remove data.

Think of it as a way to structure how systems interact using URLs and these HTTP methods. It is simple, lightweight, and widely used in web development.

**Question 2: What is Json-Server? How we use in React ?**

JSON-Server is a tool that converts a JSON file into a REST API. It is useful for front-end development when the backend is not ready.

### Usage in React:

npm install -g json-server

**1 : Create db.json: Add some sample data:**{ "users": [{ "id": 1, "name": "John", "age": 30 },

{ "id": 2, "name": "Jane", "age": 25 }] }

**2 : Start the Server:**json-server --watch db.json --port 3001

Use in React: Fetch data from http://localhost:3001 using Axios or fetch API.

**Question 3: How do you fetch data from a Json-server API in React? Explain the role of fetch() or axios() in making API requests.**

### **Fetching Data from JSON-Server in React**

**To fetch data from a JSON-server API in React, you can use fetch() or axios() to make HTTP requests.**

### **Steps:**

**Start the JSON-Server:  
json-server --watch db.json --port 3001**

**Fetch Data in React: using fetch():  
useEffect(() => {**

**fetch('http://localhost:3001/users')**

**.then(response => response.json())**

**.then(data => setUsers(data)); }, []);**

**using axios():  
useEffect(() => {**

**axios.get('http://localhost:3001/users')**

**.then(response => setUsers(response.data));}, []);**

### **Role of fetch() and axios():**

* **fetch(): A built-in JavaScript function for making HTTP requests. Returns a promise.**
* **axios(): A library for easier and more advanced HTTP requests. Supports additional features like interceptors and error handling.**

**Both are used to get or send data from/to APIs.**

**Question 4: What is Firebase? What features does Firebase offer?**

**Firebase is a Backend-as-a-Service (BaaS) platform by Google. It provides tools and services to build, improve, and scale web and mobile applications without managing backend infrastructure.**

### **Key Features of Firebase:**

1. **Authentication: User authentication using email, phone, Google, Facebook, etc.**
2. **Realtime Database: Cloud-hosted NoSQL database for real-time data synchronization.**
3. **Firestore: Scalable, flexible NoSQL database for more complex queries.**
4. **Cloud Storage: Store and serve user-generated files like images and videos.**
5. **Hosting: Fast and secure web hosting for static and dynamic content.**
6. **Cloud Functions: Run backend code in response to events or HTTP requests.**
7. **Analytics: Track user behavior and performance metrics.**
8. **Push Notifications: Send notifications to users across platforms.**

### **Usage:**

**Firebase simplifies backend development, allowing developers to focus on building the app's frontend.**

**Question 5: Discuss the importance of handling errors and loading states when working withAPIs in React**

### **Why Handle Errors and Loading States in React API Calls?**

1. **Loading State:**
   * **Show a message like "Loading..." or a spinner while waiting for data.**
   * **Helps users know the app is working, not stuck.**
2. **Error State:**
   * **Show a message like "Something went wrong!" if the API fails.Prevents the app from crashing and keeps it running smoothly.**

### **Example:**

**if (loading) return <p>Loading...</p>; // Show while waiting for data.**

**if (error) return <p>Error occurred!</p>; // Show if something goes wrong.**

**Why Important?**

* **Makes the app user-friendly.**
* **Helps developers fix issues quickly.**
* **Keeps the app stable even if the API has issues.**

**Context API**

**Question 6: What is the Context API in React? How is it used to manage global state across multiple components?**

**The Context API is a feature in React that allows you to manage global state and share data across multiple components without prop drilling (passing props through every level of the component tree).**

**Create Context:  
 Use React.createContext() to create a context object.  
  
 const MyContext = React.createContext();**

1. **Provide Context:  
    Use the Provider component to pass the global state to the component tree.  
     
    <MyContext.Provider value={/\* state \*/}>**

**<App />**

**</MyContext.Provider>**

1. **Consume Context:  
    Use useContext() hook or Context.Consumer to access the global state in any component.  
     
    const value = useContext(MyContext);**

**Question 7: Explain how createContext() and useContext() are used in React for sharing state.**

### **createContext() and useContext() in React**

1. **createContext():**
   * **Used to create a Context object that will hold the global state.**
   * **Returns a Provider and Consumer component.**

**const MyContext = React.createContext();**

1. **useContext():**
   * **A hook used to access the value provided by the Context Provider.**
   * **It allows any component to consume the context data without prop drilling.**
2. **const value = useContext(MyContext);**

**Code :**

**// Create Context**

**const MyContext = React.createContext();**

**// Provide context in a parent component**

**<MyContext.Provider value="Hello">**

**<ChildComponent />**

**</MyContext.Provider>**

**// Consume context in a child component**

**const ChildComponent = () => {**

**const value = useContext(MyContext);**

**return <p>{value}</p>; // Output: Hello**

**};**