**1)What do you mean by Restful web services?**

**RESTful services use standard HTTP methods and are designed to be stateless, scalable, and easy to use. Here are some key characteristics of RESTful web services:**

###### ****Statelessness: Each request from a client to a server must contain all the information needed to understand and process the request.****

1. **Resource-Based: REST treats data as resources, which can be identified by unique URIs (Uniform Resource Identifiers).**
2. **HTTP Methods: RESTful services use standard HTTP methods to perform operations on resources:**
3. **GET:Collection of resources**
4. **Post:create a new resource**
5. **Put:Update an existing resources**
6. **Delete:Remove a resource**
7. **What is Json-Server? How we use in React ?**

**Ans)JSON Server is a lightweight Node.js tool that allows developers to create a mock RESTful API using a JSON file as the data source. In React, you can use JSON Server to simulate backend interactions, enabling you to develop and test your application without needing a real API. How to Use JSON Server in React**

**Ex. {**

**"posts": [**

**{ "id": 1, "title": "Post 1", "content": "Content of Post 1" },**

**{ "id": 2, "title": "Post 2", "content": "Content of Post 2" }**

**]**

**}**

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

**Ans)I)To fetch data from a JSON Server API in React, you can use either the fetch() method or the Axios library.**

**Ii)The fetch() method is a built-in JavaScript function that allows you to make network requests to retrieve resources, returning a promise that resolves to the response.**

**Iii)Axios, on the other hand, is a third-party library that simplifies HTTP requests and provides additional features such as automatic JSON data transformation and request interceptors.**

**Ex. import React, { useEffect, useState } from 'react';**

**const MyComponent = () => {**

**const [data, setData] = useState([]);**

**const [loading, setLoading] = useState(true);**

**const [error, setError] = useState(null);**

**useEffect(() => {**

**fetch('http://localhost:3030/posts')**

**.then(response => {**

**if (!response.ok) {**

**throw new Error('Network response was not ok');**

**}**

**return response.json();**

**})**

**.then(data => {**

**setData(data);**

**setLoading(false);**

**})**

**.catch(error => {**

**setError(error);**

**setLoading(false);**

**});**

**}, []);**

**if (loading) return <div>Loading...</div>;**

**if (error) return <div>Error: {error.message}</div>;**

**return (**

**<ul>**

**{data.map(item => (**

**<li key={item.id}>{item.title}</li>**

**))}**

**</ul>**

**);**

**};**

**export default MyComponent;**

1. **What is Firebase? What features does Firebase offer?**

**Ans)Firebase is a platform developed by Google that provides a variety of tools and services to help developers build high-quality applications.**

1. **Real time database: NoSQL cloud database that allows data to be stored and synchronized in real-time across all clients**
2. **Authentication:Provides easy-to-use authentication services, including email/password, social media logins, and anonymous authentication**
3. **Cloud firestore:** **flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform**
4. : **Discuss the importance of handling errors and loading states when working withAPIs in React**

**Ans) I)User experience: Loading Feedback: When you fetch data, showing a loading message or spinner lets users know that something is happening. It keeps them informed and prevents confusion**

**Error Messages: If something goes wrong (like a network issue), showing a clear error message helps users understand that there was a problem**

**ii)Debugging:Easier to Fix Issues: When you catch errors and log them, it’s easier for developers to find and fix problem**

**Iii)Performance optimization:Better Performance: By managing loading states, you can prevent unnecessary updates to the screen, making your app run smoother**