**JSON-server and Firebase Real Time**

**Database**

1. What do you mean by RESTful web services?

* **RESTful Web Services** are web services that follow the **REST (Representational State Transfer)** architectural style.
* REST is a set of constraints and principles for building web services that allow communication between different systems over the internet.
* **Key Features of RESTful Web Services.**

**1.Stateless**

* **Each request from a client contains all the necessary information, and the server does not store client state between requests.**

**2.Client-Server Architecture**

* **The client and server are independent of each other, allowing flexibility in development and scalability.**

**3. Use of HTTP Methods**

**GET → Retrieve data**

**POST → Create a new resource**

**PUT → Update an existing resource**

**DELETE → Remove a resource**

1. What is Json-Server? How we use in React.

* **JSON-Server** is a lightweight, full-fledged **mock REST API** that allows developers to create a **fake backend** using a simple **JSON file**.
* It helps simulate a real server without writing backend code, making it useful for prototyping, testing, or frontend development.
* JSON-Server is a powerful tool to create a **mock REST API** in seconds.
  1. **No backend required** – Just use a JSON file as a database.
  2. **REST API Ready** – Supports standard HTTP methods (**GET, POST, PUT, DELETE, PATCH**).
  3. **Fast setup** – Can be installed and run in seconds.
  4. **Supports Query Parameters** – Easily filter, paginate, and search data.

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

* To fetch data from a **JSON-Server API** in **React**, you can use either fetch() or **Axios**. Both methods allow you to make **HTTP requests** to retrieve or manipulate data from an API.
* Using fetch() to Get Data
* The fetch() function is a built-in JavaScript method used for making **network requests**. It **returns a Promise** that resolves to the response of the request.
* Built-in method in JavaScript.
* Supports **GET, POST, PUT, DELETE** requests.
* Using Axios to Fetch Data
* Axios is a **popular third-party library** for making API requests. It provides a **simpler syntax x** than fetch(), supports **automatic JSON transformation**, and better **error handling**.
* Simpler syntax than fetch().
* Automatically transforms **JSON response**.
* **Error handling** is better than fetch()

4.What is Firebase? What features does Firebase offer?

* Firebase is a **Backend-as-a-Service (BaaS)** platform provided by **Google** that helps developers build web and mobile applications **without managing servers**.
* It offers a variety of **backend services**, including **authentication, real-time databases, cloud storage, hosting, and analytics**.
* Key Features of Firebase.
  1. Authentication (Firebase Auth)
  2. Real-time Database (Firebase Realtime Database)
  3. Firestore (Cloud Firestore)
  4. Firebase Cloud Storage
  5. Firebase Hosting
  6. Firebase Cloud Messaging (FCM)
  7. Firebase Analytics
  8. Firebase Remote Config
  9. Firebase Crashlytics
  10. Firebase Cloud Functions
* Why Use Firebase

**No need to manage servers** (Backend-as-a-Service).

**Scalable & real-time data syncing**.

**Built-in authentication & security**.

**Free tier available** (with limitations).

5.Discuss the importance of handling errors and loading states when working with APIs in React.

* When working with **APIs** in React, handling **loading states** and **errors** is essential for creating a **seamless and user-friendly experience**.
* Without proper handling, users might face **blank screens, infinite loaders, or broken UIs** when an API call fails or takes time to respond
* Why Handle Loading States?

When fetching data from an API, there is a delay between sending the request and receiving the response. **Loading states** ensure users see a proper **"loading" indicator** instead of an empty UI.

* Issues Without Handling Loading States

1. Users may see a **blank screen**, thinking the app is broken.
2. Users might click buttons multiple times, causing **duplicate API calls**.
3. On slow internet, the user **has no feedback** that data is being fetched.

* Best Practices for Loading States
* Use a **state variable (isLoading)** to track loading.
* Show a **spinner, skeleton loader, or message** while fetching data.