❖ React – JSON Server & Firebase Real Time Database

1. What do you mean by RESTful web services?

- RESTful web services are web services that use the Representational State Transfer (REST) architectural style.
- This style designed to make the best use of the HTTP protocol, which is the language of the internet.

2. What is Json-server? How we use in React?

- JSON-Server is a simple and lightweight fake REST API server that you can use to mock a backend while developing front-end applications.
- It allows you to create a mock API using a db.json file without setting up a full-fledged server.

How to use JSON-Server in React: -

- Step 1 – Install JSON-Server → [npm install json-server]
- Step 2 – Create db.json file.
- Step 3 – Start the JSON-Server.
- Step 4 – Use the JSON-Server in React.

3. 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 JSON-Server API in React, you can use either the fetch () API or the axios library.
- Both tools allow you to make HTTP requests from your React Components to retrieve or send data to the server.
- **Role of fetch ()** is Built in JavaScript method. And Returns a promise that needs to be manually converted to JSON using. json ().
- Requires manual handling with. catch ().
- Basic HTTP methods support.
- **Role of axios ()** is Third party library. And Automatically parses JSON by default.
- Easily configurable with default settings.
- Supports older browsers.

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

- Firebase is a Backend-as-a-Service (BaaS) platform developed by Google that provides a suite of cloud-based tools and services for building and managing web and mobile applications without the need for managing servers.
- It helps developers focus on building their applications while Firebase handles backend infrastructure such as databases, authentication, file storage, and more.

Features of Firebase to offer: -

1. Firebase Authentication
2. Firebase Real-Time Database
3. Cloud Firestore
4. Firebase Hosting
5. Firebase Cloud Storage
6. Firebase Crashlytics
7. Firebase Analytics
8. Firebase Cloud Messaging
9. Firebase Performance Monitoring
10. Firebase Remote Config
11. Firebase Test Lab

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

- When working with APIs in React (or any front-end application), handling **errors** and **loading states** is essential to ensure a smooth and user-friendly experience.

1. Loading States: -

    - When making an API call, there is often a delay between the request and the response. During this delay, users need to be aware that the application is processing their request.
    - Enhance User Experience
    - Prevents User Confusion
    - Maintain UI Responsiveness
    - Reduce User Errors

2. Error Handling: -

- APIs are prone to various issues such as network failures, server errors, and invalid responses. Handling these errors gracefully is crucial.
- Prevents Application Crashes
- Provides Clear Communication
- Support Debugging and maintenance