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

Answer:
RESTful web services are APIs that follow the principles of REST (Representational State Transfer). They use standard HTTP methods (GET, POST, PUT, DELETE) to manage resources on a server. REST APIs are stateless, meaning each request contains all the necessary information to complete it, and responses are usually in formats like JSON or XML.

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

Answer:
JSON Server is a tool to create a mock REST API from a JSON file for testing purposes. In React:

1. Install it using `npm install json-server --save-dev`.
2. Create a `db.json` file with mock data.
3. Start the server with `json-server --watch db.json --port 5000`.
4. Make API requests from your React app to `http://localhost:5000`.

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.

Answer:
You can use `fetch()` or `axios()` to fetch data from a JSON Server API:

- **`fetch()`**: A built-in JavaScript function to make HTTP requests. Example:

  ```
  js
  CopyEdit
  fetch('http://localhost:5000/items')
  .then(response => response.json())
  .then(data => setItems(data));
  ```

- **`axios()`**: An external library that simplifies making requests, with added features like better error handling. Example:

  ```
  js
  CopyEdit
  import axios from 'axios';
  axios.get('http://localhost:5000/items')
  .then(response => setItems(response.data));
  ```

Both are used to send requests and handle responses from an API.

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

Answer:
Firebase is a Google platform for building mobile and web apps, offering features like:

- **Realtime Database**: Syncs data in real-time.
- **Authentication**: Allows users to sign in with email, social media, etc.
- **Hosting**: Provides fast and secure hosting for web apps.
- **Cloud Functions**: Lets you run server-side code.
- **Firestore**: A scalable database for storing data.
- **Push Notifications**: Send notifications using Firebase Cloud Messaging.

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

Answer:
Handling loading and error states is essential for a smooth user experience:

- **Loading States**: Inform users that data is being fetched (e.g., showing a loading spinner).
- **Error Handling**: Display error messages when something goes wrong (e.g., network issues).