**React – JSON-server and Firebase Real Time Database**

Q1: What do you mean by RESTful web services?

Ans: - RESTful web services are web APIs that follow the principles of REST (Representational State Transfer), an architectural style that uses standard HTTP methods like GET, POST, PUT, and DELETE to perform CRUD operations on resources, which are typically represented in JSON or XML format. These services are stateless, scalable, and commonly used to allow communication between client and server in modern web applications.

Q2: What is Json-Server? How we use in React ?

Ans: - JSON-Server is a simple, lightweight fake REST API server that lets developers mock backend data quickly using a db.json file. In React, it is commonly used for prototyping or testing by running json-server --watch db.json, which provides endpoints like a real backend, and React apps can then fetch or manipulate data via HTTP methods.

Q3: How do you fetch data from a Json-server API in React? Explain the role of fetch() or axios() in making API requests.

Ans: To fetch data from a JSON-server API in React, we typically use fetch() or axios() inside a useEffect() hook to make asynchronous API calls. fetch() is a built-in JavaScript method to make HTTP requests, while axios() is a third-party library that provides a simpler syntax and automatic JSON parsing. Both allow React apps to retrieve or modify data from the API and update the component state accordingly.

Q4: What is Firebase? What features does Firebase offer?

Ans: Firebase is a Backend-as-a-Service (BaaS) platform by Google that provides various tools for building and scaling web and mobile applications. Its key features include real-time NoSQL database (Firestore), authentication, cloud functions, hosting, storage, and analytics, making it easy for developers to implement backend functionalities without managing servers.

Q5: Discuss the importance of handling errors and loading states when working with APIs in React.

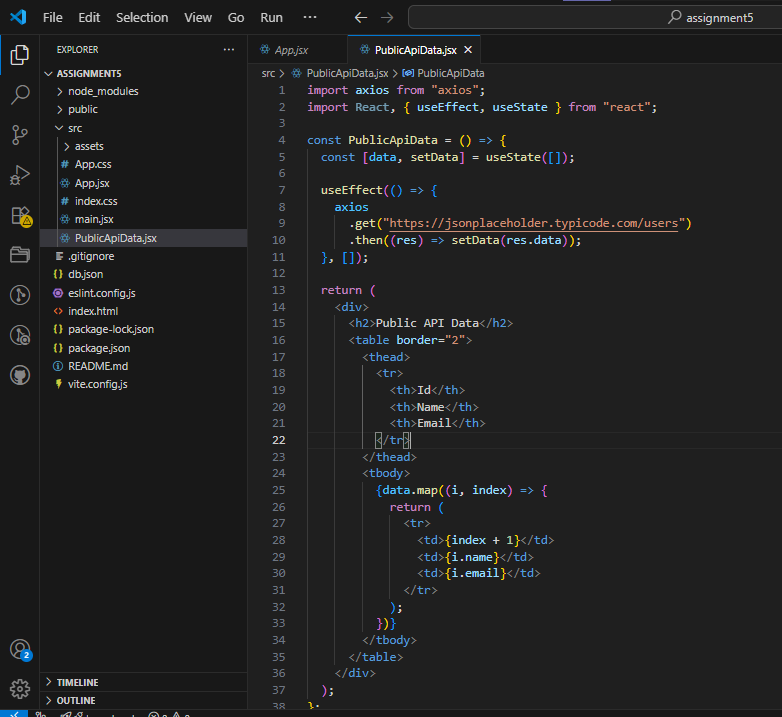
Ans: Handling errors and loading states is crucial when working with APIs in React to provide a good user experience and ensure app stability. Without proper loading indicators, users may be confused during network delays, and unhandled errors (like failed requests) can break the UI or result in incorrect data being shown. Using conditional rendering and try-catch or .catch() blocks helps manage these states effectively.

**Lab Tasks**

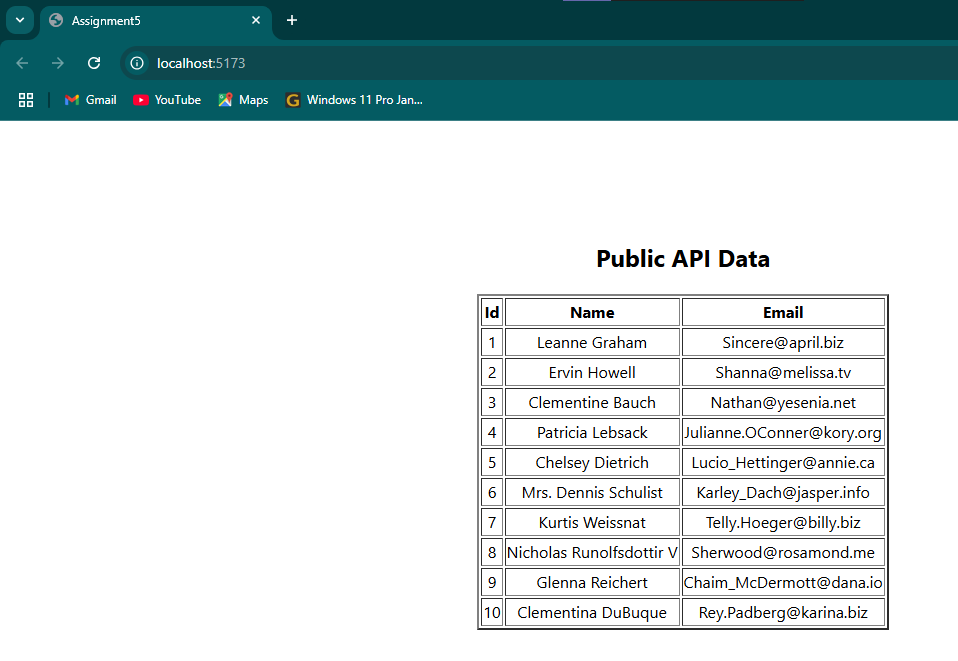
**Task 1 :**

* **Create a React component that fetches data from a public API (e.g., a list of users) and displays it in a table format.**

**Code:**

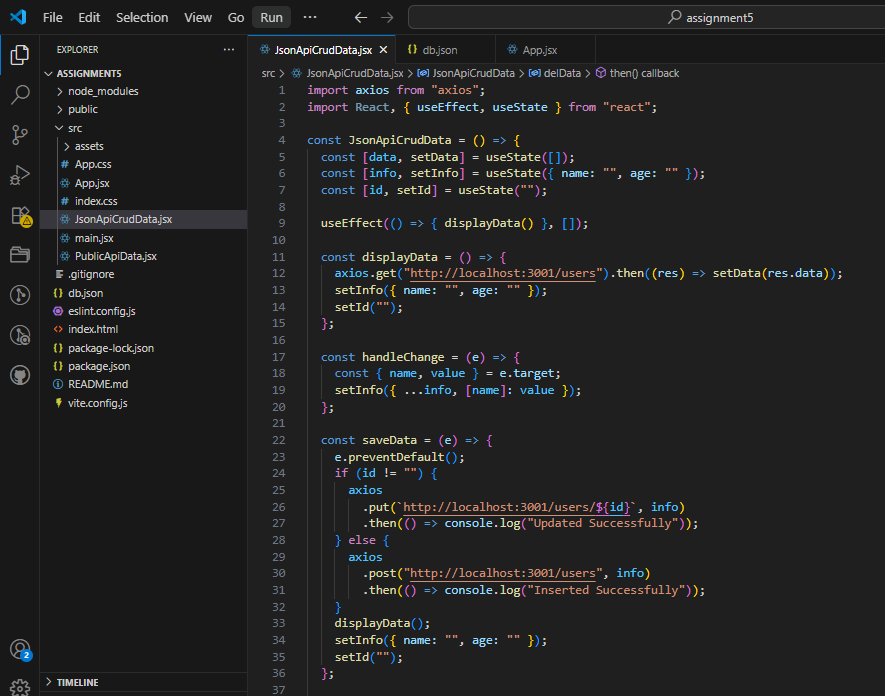
****

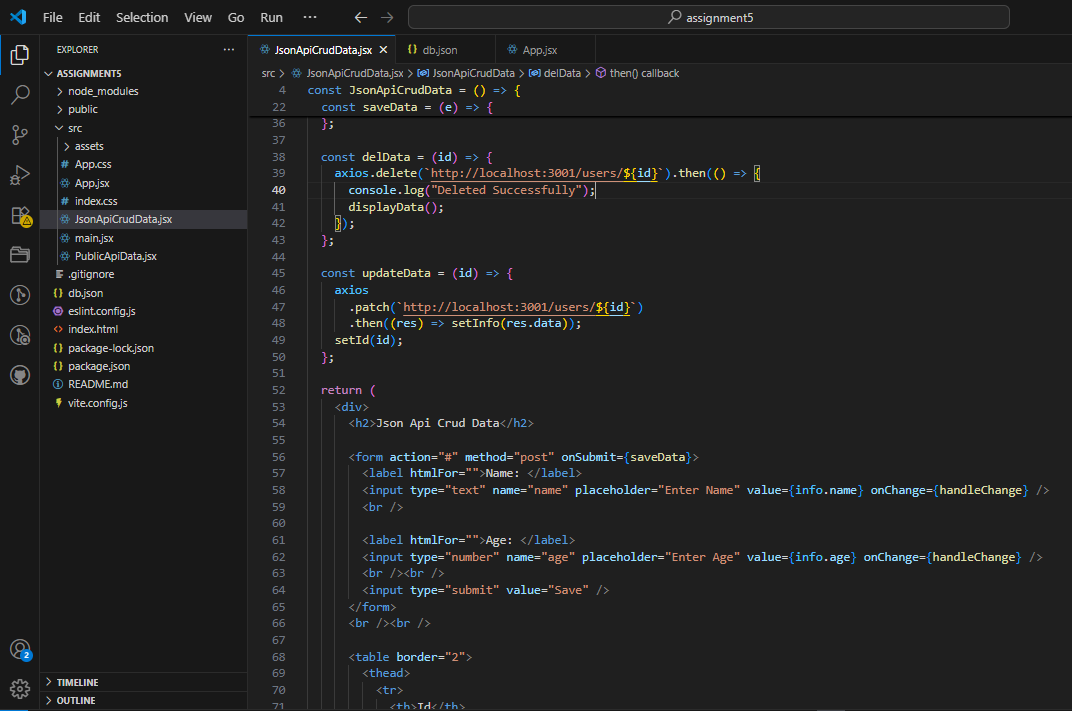
**Output:**

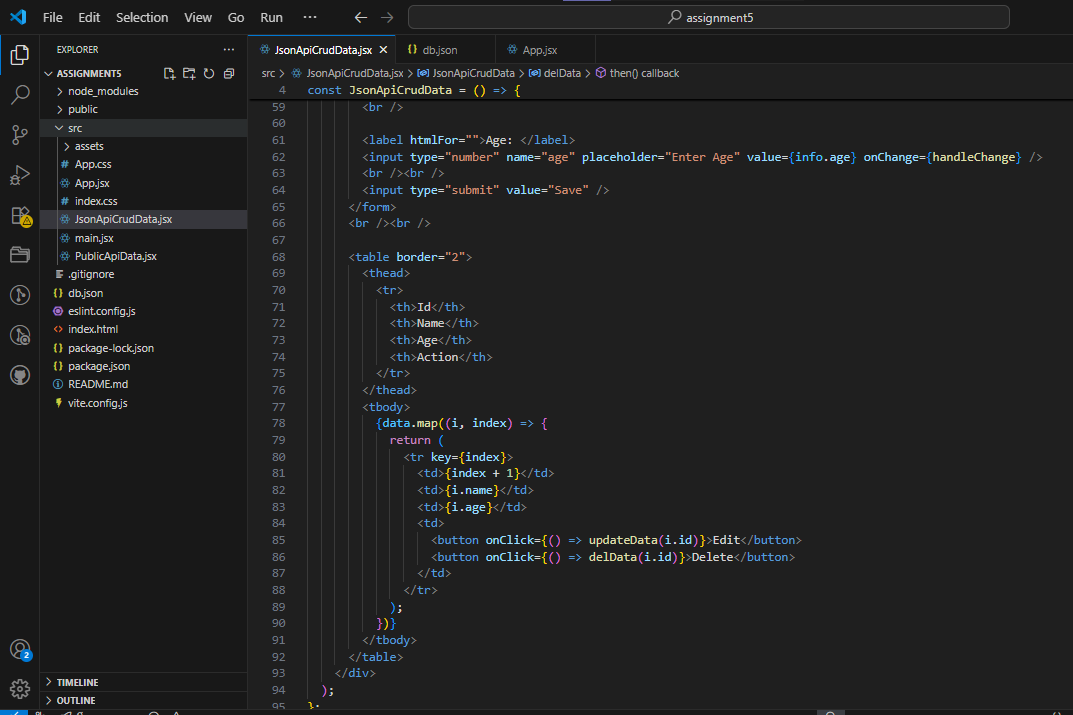
****

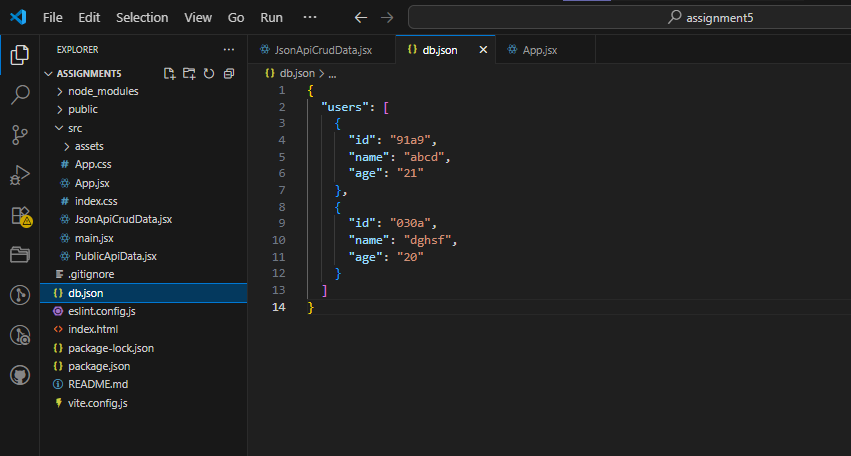
* **Create a React app with Json-server and use Get , Post , Put , Delete & patch method on Json-server API.**

**Code:**

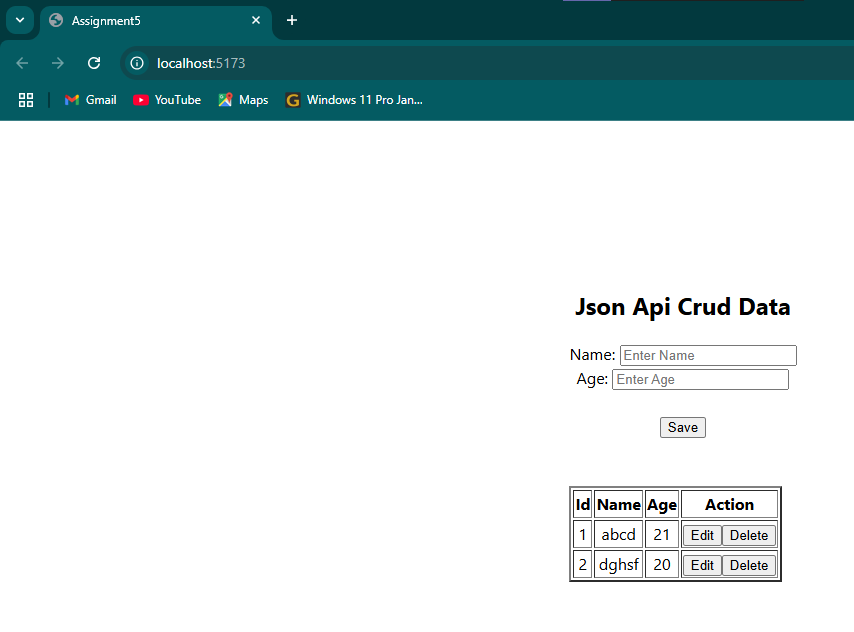
****

****

****

****

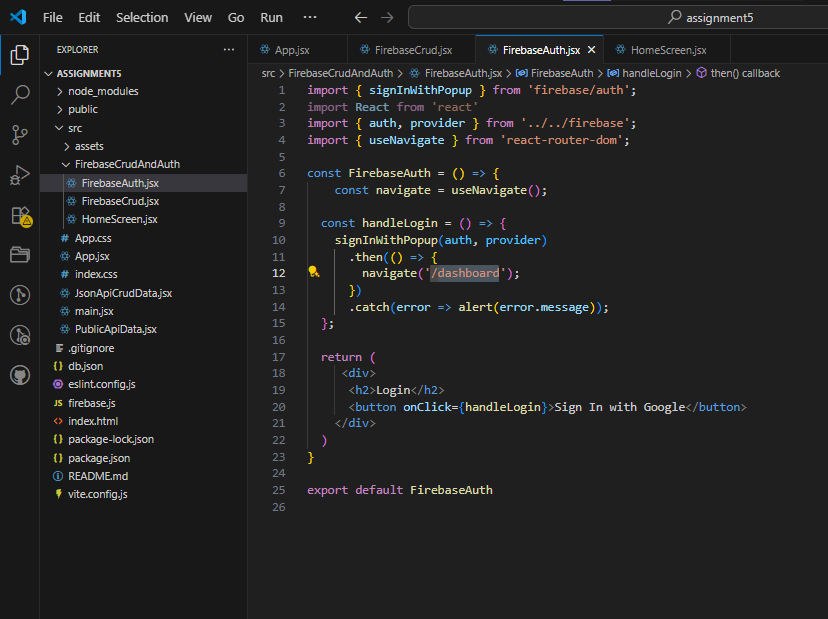
**Output:**

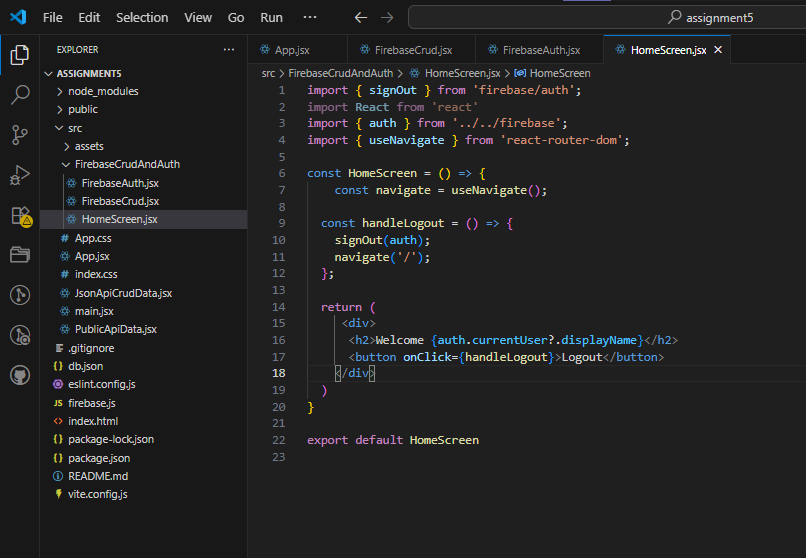
****

**Task 2 :**

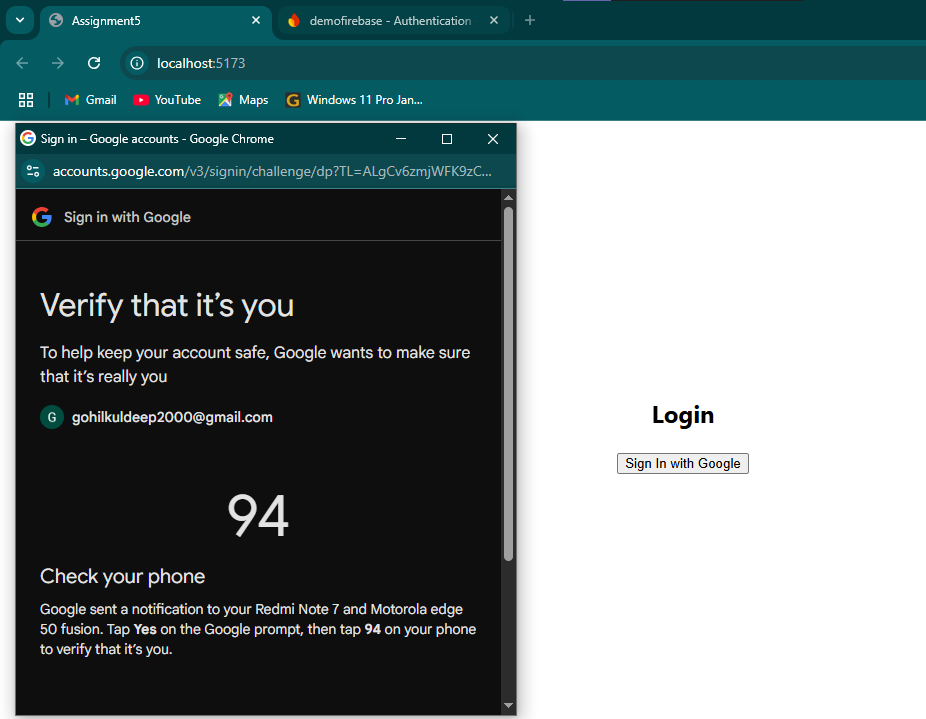
* **Create a React app crud and Authentication with firebase API.**
* **Implement google Authentication with firebase API.**

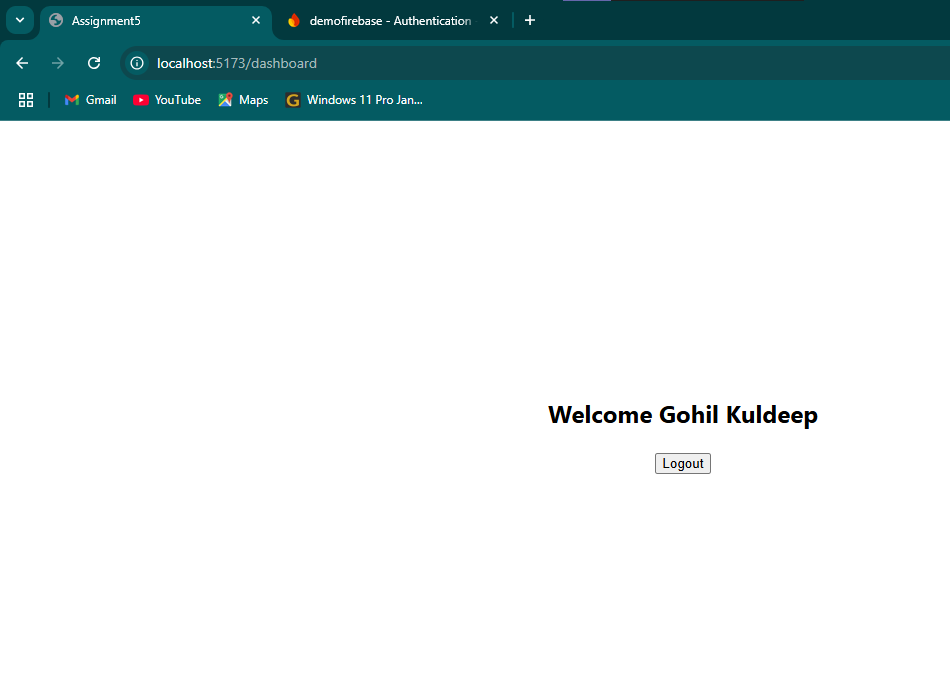
**Code:**

****

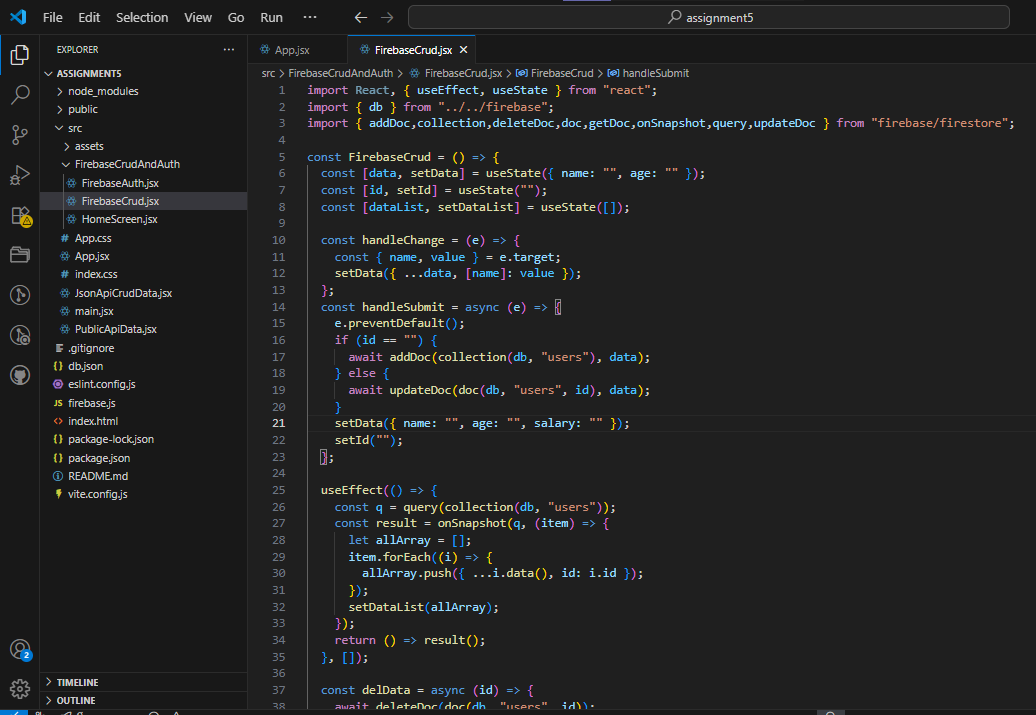
****

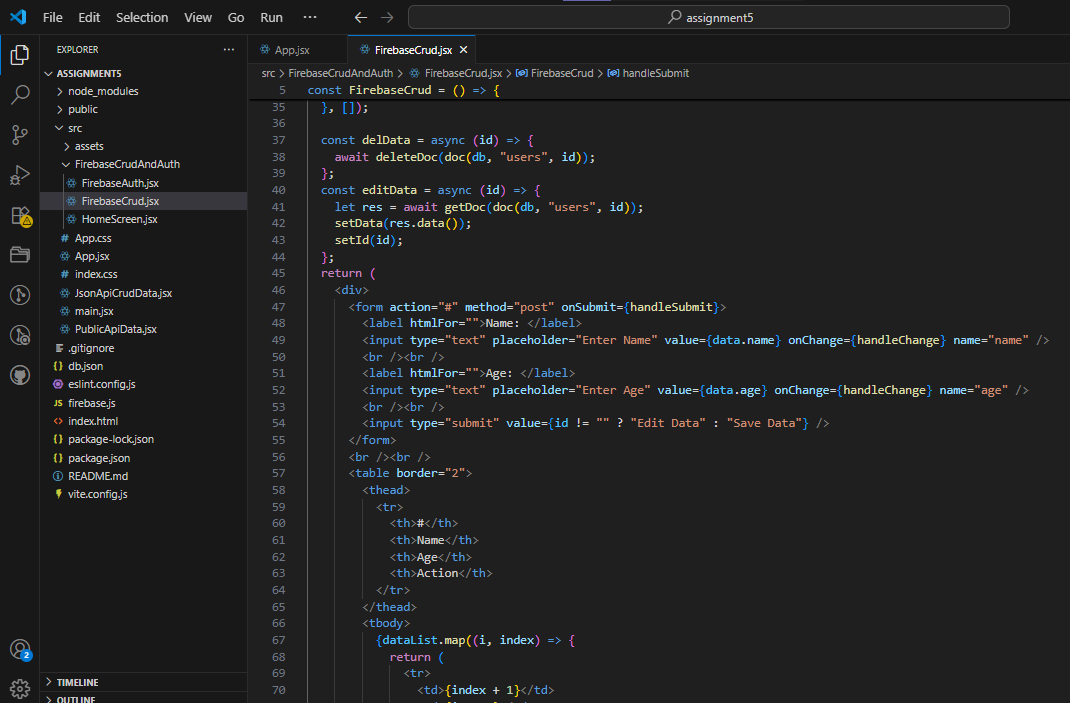
**Output:**

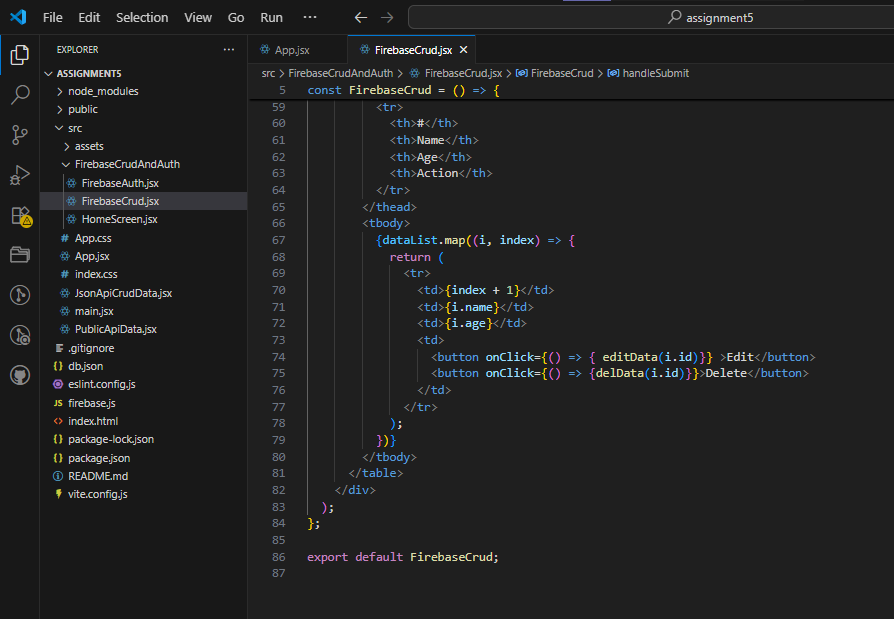
****

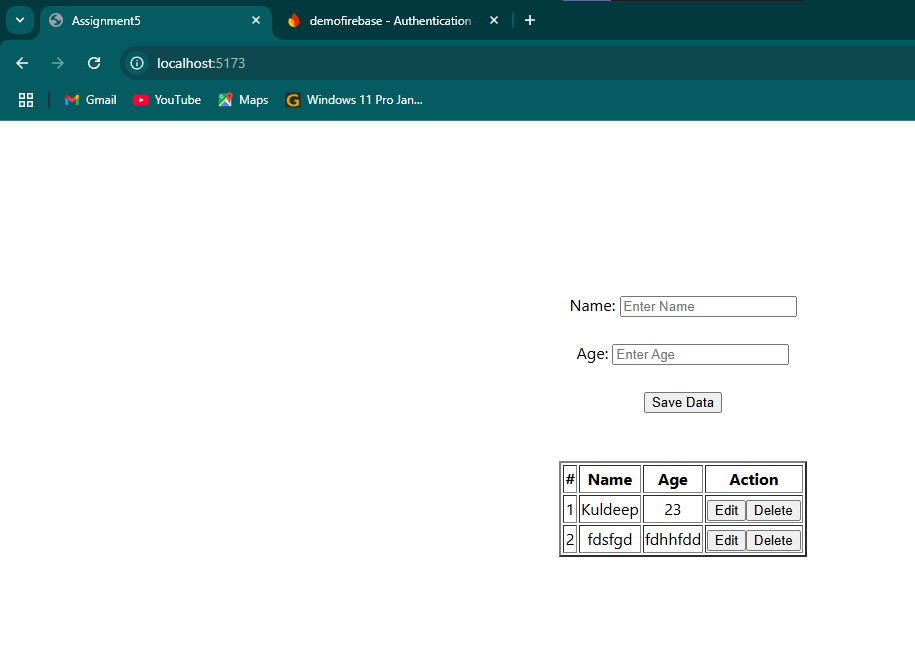
****

**Code:**

****

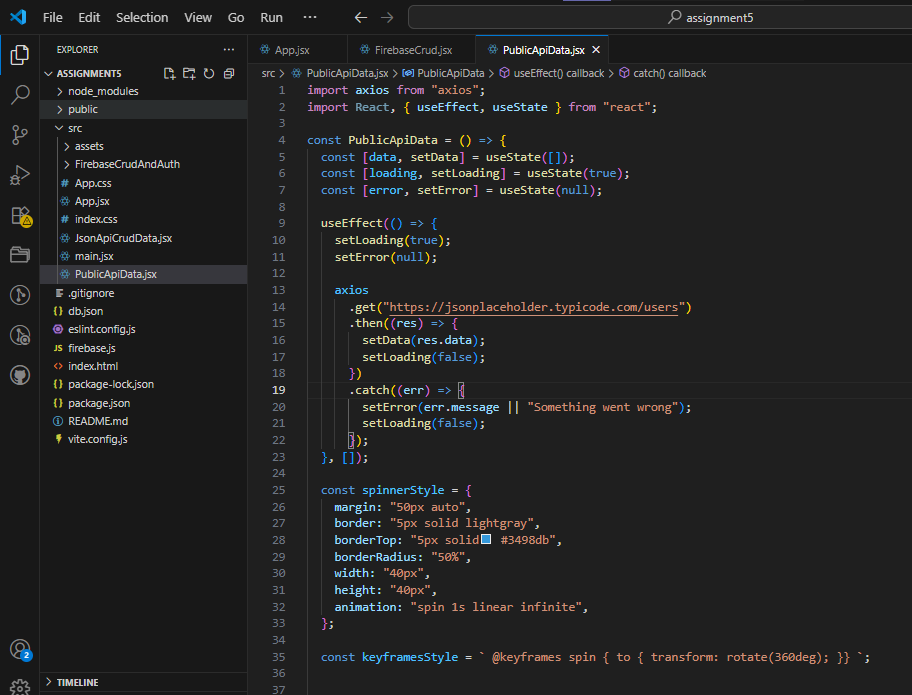
****

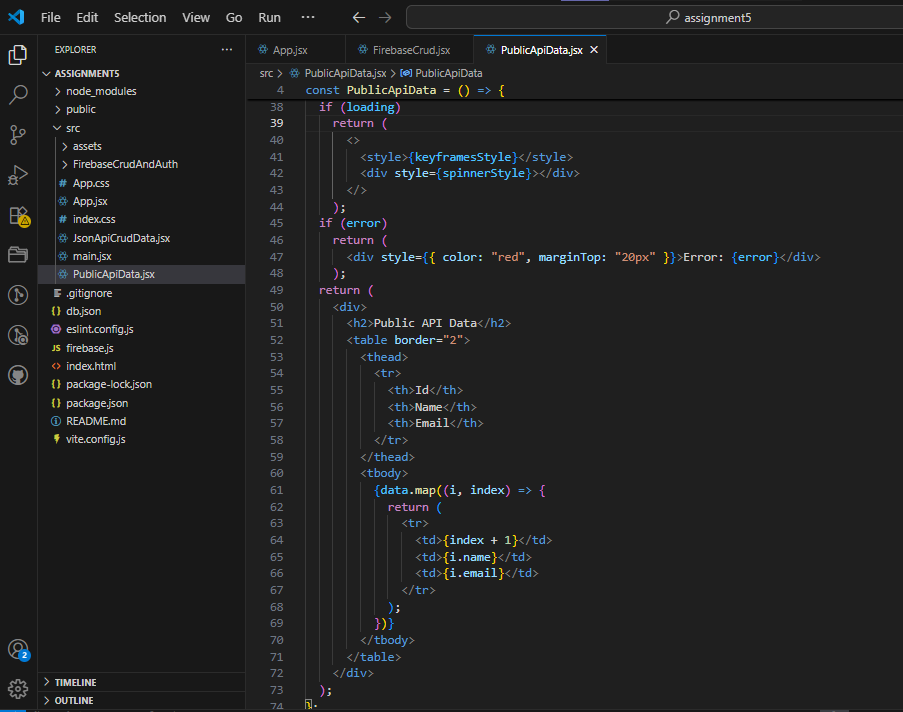
****

****

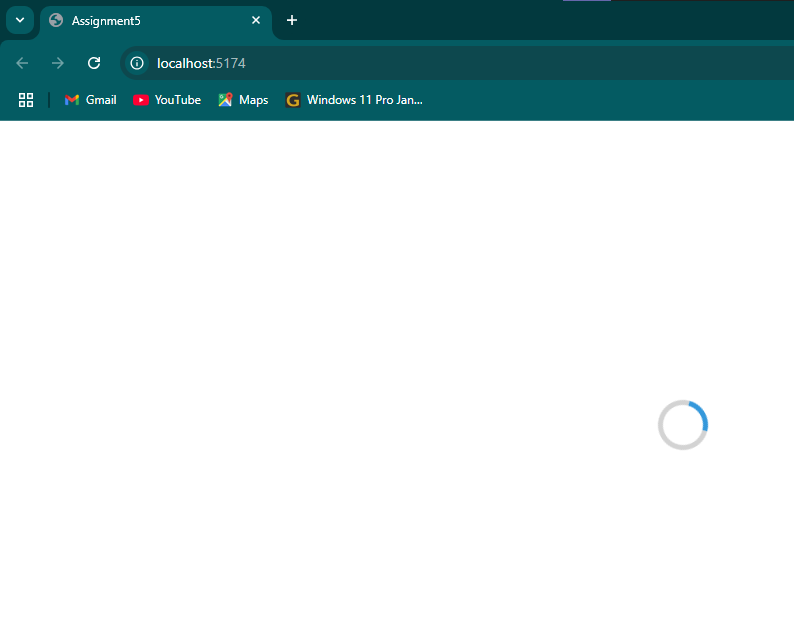
**Task 2 : Implement error handling and loading states for the API call. Display a loading spinner while the data is being fetched.**

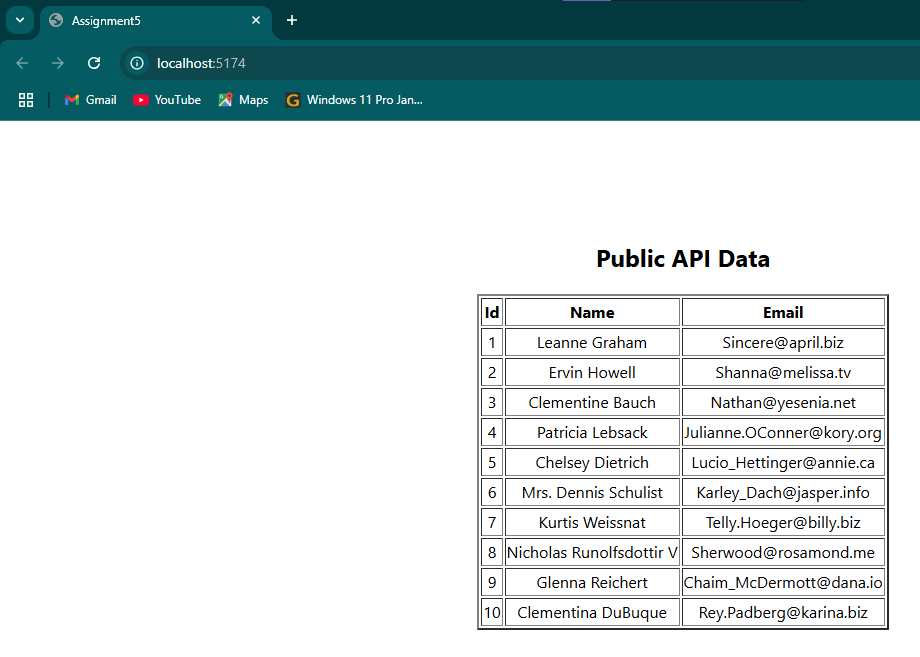
**Code:**

****

****

**Output:**

****

****

**Context API**

Q1: What is the Context API in React? How is it used to manage global state across multiple components?

Ans: - RESTful web services are web APIs that follow the principles of REST (Representational State Transfer), an architectural style that uses standard HTTP methods like GET, POST, PUT, and DELETE to perform CRUD operations on resources, which are typically represented in JSON or XML format. These services are stateless, scalable, and commonly used to allow communication between client and server in modern web applications.