Next Steps: Integrating a Mock Backend API with Your React App

**Objective:**  
Take your existing React application that uses a third-party API and extend it by creating additional components that consume data from a mock backend API using JSON Server. Eventually, we will transition to using an API Gateway.

**Instructions**

1. **Review Your Existing React App**  
   You’ve already created a React application that consumes data from a third-party API. Now, we’ll extend this application to work with a custom backend API.
2. **Set Up JSON Server to Mock a Backend API**  
   JSON Server is a tool that allows you to create a full fake REST API with zero coding. We’ll use it to mock our backend API for now.
   1. **Install JSON Server**  
      In your project directory, install JSON Server: npm install -g json-server
   2. **Create a Mock Data File**  
      Create a db.json file in your project root and define your mock data. Here’s an example:

{

"posts": [

{ "id": 1, "title": "Hello World", "content": "This is a post." }

],

"comments": [

{ "id": 1, "postId": 1, "content": "Nice post!" }

]

}

* 1. **Start JSON Server**  
     Start the server to serve your mock data: json-server --watch db.json --port 5000

1. **Create Additional Components in Your React App**  
   Extend your React app by creating new components that consume data from your mock API. For example:

* PostsList: Fetch and display a list of posts.
* PostDetail: Display detailed information about a single post.
* CommentsList: Fetch and display comments for a specific post.

1. **Use Axios or Fetch to Interact with the API**  
   Use Axios or the Fetch API to make HTTP requests to your JSON Server API. For example, to fetch posts