Here’s a complete **React + Axios + Dummy API** assignment that covers:

✅ CRUD  
✅ Login & Auth  
✅ Protected Route  
✅ Recipe Listing with Pagination & Search  
✅ Add/Edit/Delete  
✅ Testing with Jest  
✅ Bonus: Axios Interceptor

**🍲 Assignment: Recipe Manager (with DummyJSON)**

**🔧 Backend API:** [**https://dummyjson.com**](https://dummyjson.com/)

📌 You will use **https://dummyjson.com/recipes** as the core endpoint.

**🎯 Objective**

Build a recipe manager where users can:

* Log in
* View a list of recipes (with pagination + search)
* Add, edit, delete recipes (only if logged in)
* Write unit tests for components and API logic

**🧩 Requirements**

**🔐 1. Login Page**

* Use this API to login: POST https://dummyjson.com/auth/login

{

"username": "kminchelle",

"password": "0lelplR"

}

* On success:
  + Store accessToken in localStorage
  + Redirect to /recipes

**📜 2. Recipe Listing Page**

* Fetch recipes from:
* GET https://dummyjson.com/recipes?limit=10&skip=0
* Implement:
  + Pagination
  + Search by name/title
* Display:
  + Title
  + Cuisine
  + Prep time
  + Calories

**➕ 3. Add Recipe Page (protected)**

* Only accessible when logged in
* Use:
* POST https://dummyjson.com/recipes/add
* Form Fields:
  + Name
  + Cuisine
  + Ingredients
  + Instructions
* After submission, show success message and redirect to /recipes

**✏️ 4. Edit / Delete Recipes**

* Edit:
* PUT https://dummyjson.com/recipes/{id}
* Delete:
* DELETE https://dummyjson.com/recipes/{id}

**🔒 5. Protected Routes**

* Prevent access to /add, /edit, or /delete unless user is logged in
* Use React Router + useNavigate + localStorage

**🧪 6. Testing (with Jest + React Testing Library)**

Write unit tests for:

* Login logic
* Recipe list renders items correctly
* AddRecipeForm submits correctly with mock API
* axios interceptor adds token header

**🛠 Tools Suggested**

* React
* Axios
* React Router
* React Hook Form (optional)
* Tailwind / Bootstrap (optional)
* React Testing Library + Jest

**📝 Bonus**

* Use an **Axios interceptor** to attach Authorization token
* Use useContext or Redux to manage auth state
* Add a loading spinner on API calls

**📂 Suggested File Structure**

/src

/pages

- Login.tsx

- RecipeList.tsx

- AddRecipe.tsx

- EditRecipe.tsx

/components

- RecipeCard.tsx

- Pagination.tsx

/services

- recipeService.ts

- authService.ts

- axiosConfig.ts

/tests

- Login.test.tsx

- RecipeList.test.tsx

- recipeService.test.ts

/context

- AuthContext.tsx